FME Server 2010 - Web Services API Reference

概览

本文档可为要利用 FME Server Web 服务 API 构建 Web 应用程序的开发者提供参考。本文档介绍了 FME Server 开发 Web 服务的规范。它包含 HTTP 请求和响应的规则，以及每个服务的请求和响应元素。

FME Server 服务

FME Server 包含以下 Web 服务:

- Data Download Service（数据下载服务）
- Data Streaming Service（数据流服务）
- Data Upload Service（数据上传服务）
- Job Submitter Service（任务提交服务）
- KML Network Link Service（KML 网络链接服务）
- OGC Web Feature Service（OGC Web 要素服务）

- OGC Web Map Service（OGC Web 地图服务）

基本的服务组成要素

- General Request Rules
  - HTTP GET
  - HTTP POST
  - Request Parameters Common to All Services
    - Workspace Published Parameters
    - Transformation Manager Directives
    - FME Engine Directives
- General Response Rules
  - XML Response Schema
- HTTPS
- Authentication

所有服务共同的请求参数

- Workspace Published Parameters（工作区发布参数）
- Transformation Manager Directives（转换管理指令）
- FME Engine Directives（工作区发布参数）

工作区发布参数

一个在工作区中定义的 FME 发布参数可以直接传递到 Web 服务作为一个请求参数。当 Web 服务提交一个转换任务到 FME Server，FME Server 使用该发布参数值来运行工作区。

语法（Syntax）

通过使用 HTTP POST 或 GET 方法，发布参数可以提交到工作区。假设使用了 GET 方法，那么使用发布参数、URL 参数的语法如下：

```
http://hostname/<service>/<repository>/<workspace>?{ParameterName[=ParameterValue]&}
```

范例

这个网址通过数据下载服务使用 austinDownload.fmw 工作区（请注意 URL 参数，如 COORDSYS，它的值 LL84）：

```
http://hostname/fmedatadownload/Samples/austinDownload.fmw?COORDSYS=LL84&BBOX_COORDSYS=LL84
&FORMAT_GENERIC=SHAPE&MINX=-100&MINY=25&MAXX=-90&MAXY=35&THEMES=airports
```

以上示例中发布参数的论述见下表。 您也可以从示例库中下载 austinDownload.fmw 工作区，以了解如何在工作区中配置发布参数。

参数名称	参数值	描述
COORDSYS	LL84	这是 Writer 的*坐标系统*参数（即输出坐标）。
BBOX_COORDSYS	LL84	这是 Reader 的查找包络边界坐标系参数，查找包络边界参数由 MIN*/MAX*发布参数指定，该指定的搜索包络边界总体上会限制 FME 实际读取的数据量。此外，此发布参数用于 reproject 搜索包络边界以匹配源数据的坐标系统，以在工作区中使用 *Clipper*transformer 执行一个 hard clip。
FORMAT_GEN	SHAPE	这是一般 Writer 的*输出格式*参数。

ERIC		
MINX	-100	这是 Reader 的查找包络边界最小 *x* 参数，将用于选择和剪辑数据。Reader 与 a*Clipper* transformer 都用该发布参数来限制数据量。
MINY	25	这是 Reader 的查找包络边界最小 *Y* 参数，详述参见 MINX
MAXX	-90	这是 Reader 的查找包络边界最大 *X* 参数，详述参见 MINX
MAXY	35	这是 Reader 的查找包络边界最大 *Y* 参数，详述参见 MINX
THEMES	airport	This is the Reader's *Feature Types to Read* parameter.

转换管理指令

下表提供了有关转换管理指令的信息。

指令名	指令值	描述
tm_complexity	floating number	一个浮动的数字，表示请求的计算复杂性。
tm_description	string	一个字符串（如果它包含在双引号中的空格），表示一个请求的任意描述。
tm_fmeHost	string	一个标识主机名的字符串（没有空格），处理该请求的 FME Engine 实例正运行在该主机上。
tm_fmeInstance	string	一个标识处理请求的 FME Engine 实例的字符串（没有空格）。
tm_outputFormatTag	string	输出格式标签是 FME Server 转换管理器用于在 FME Engine 之间分配任务。当 FME Server 收到新的任务后，FME Server 转换管理器使用输出格式标签（如果指定了）将改任务请求重定向至已配置为接受该指定输出格式标签的 FME Engines。
tm_priority	integer	一个整数，表示请求的优先级。 较低的值意味着更高的优先级。优先级值必须大于或等于零。

tm_timeRequested	string	带年月日时分(yyyy-mm-dd-hh-mm) 格式的字符串。 此字符串指定 FME Server 处理该请求的时间。

FME Engine 指令

下表提供了有关 FME Engine 指令的信息。

指令名	指令值	描述
fme_LOG_FILENAME	FME translation log file name.	指定 FME 转换日志的文件名称而不是使用默认文件名。

数据下载服务

数据下载服务接受并执行指定的工作区转换的请求。如果转换成功，它返回一个 HTTP 网址供请求者下载或发送具有电子邮件内容网址的电子邮件给请求者。

服务请求的组成要素

该服务请求的 URL 地址

```
http://host[:port]/fmedatadownload/<repository>/<workspace>?{name[=value]&}
```

所有服务共同的请求参数

- Workspace Published Parameters
- Transformation Manager Directives
- FME Engine Directives

该服务的具体请求参数

下表描述了该服务的具体可选参数项.

Name	Value	Description
opt_responseformat	xml | json | html	对服务的响应格式
opt_servicemode	sync | async	在同步和异步服务模式之间切换
opt_requesteremail	comma-separated, e-mail addresses	发送通知电子邮件的地址

服务响应的组成要素

如果响应格式被指定为 XML 或 JSON，服务响应可能包含如下表所示的内容：：

Element	Child Elements	Value	描述
Status Info	message	message string	服务失败消息
	status	success | failure	服务状态
	mode	sync | async	服务模式
Fme Trans formati on Result	fmeServerResponse	FME Server response properti es	FME Server 响应
	fmeEngineResponse	FME Engine response properti es	FME Engine 响应
email	none	Email addresse s	请求者的电子邮件地址
url	none	URL string	供下载结果集（邮编）使用的 URL

jobID	none	job ID	当前任务标识符

范例

范例 1

展示了一个有两个出版参数以及指定 XML 作为响应格式的数据下载服务请求。

请求

```
http://localhost/fmedatadownload/Samples/austinDownload.fmw?THEMES=airports&FORMAT_GENERIC=SHAPE&opt_responseformat=xml
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <!-- Service Status Info -->
  <statusInfo>
    <status>success</status>
    <mode>sync</mode>
  </statusInfo>
  <!-- Service Return -->
  <fmeTransformationResult>
    <fmeEngineResponse>...</fmeEngineResponse>
    <fmeServerResponse>...</fmeServerResponse>
  </fmeTransformationResult>
  <url>http://localhost/fmedatadownloadresults/FME_3368_1245953381697_3368871561115.zip</url>
  <jobID>52</jobID>
</serviceResponse>
```

范例 2

展示了带有一个发布的参数，一个 TM 指令并指定 JSON 为响应格式的数据下载服务请求。

请求

```
http://localhost/fmedatadownload/Samples/austinDownload.fmw?THEMES=airports&tm_priority=50&opt_responseformat=json
```

响应

```
{"serviceResponse": {
  "fmeTransformationResult": {
    "fmeEngineResponse": {
      ...
    },
    "fmeServerResponse": {
      ...
   },
   "jobID": 8,
   "statusInfo": {
     "mode": "sync",
     "status": "success"
   },
   "url":
"http://localhost/fmedatadownloadresults/FME_5516_1258064613849
_5516859449883.zip"
}}
```

范例 3

展示了一个参数带有请求者的 e - mail 地址，服务为异步模式，以及指定 XML 作为响应格式的数据下载服务请求。

请求

```
http://localhost/fmedatadownload/Samples/austinDownload.fmw?opt_servicemode=async&opt_requesteremail=joe@safe.com&opt_responseformat=xml
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<serviceResponse>
  <statusInfo>
    <mode>async</mode>
    <status>success</status>
  </statusInfo>
  <email>someone@safe.com</email>
  <jobID>9</jobID>
</serviceResponse>
```

数据流服务

该服务接受并执行由一个工作区指定的转换请求，并将转换结果返回为一个数据流。

服务请求的组成要素

该服务请求的 URL 地址

```
http://host[:port]/fmedatastreaming/<repository>/<workspace>?{name[=value]&}
```

所有服务共同的请求参数

- Workspace Published Parameters
- Transformation Manager Directives

- FME Engine Directives

该服务的具体请求参数

下表描述了该服务的具体可选参数项。

Name	Value	Description
opt_responseformat	xml | json | html	该服务的响应格式

当该服务执行成功时，服务以数据流形式返回结果数据集从而 opt_responseformat 参数将被忽略； 当服务执行失败时，此参数指定将要返回的响应格式：XML，JSON 或 HTML。

服务响应的组成要素

如果响应格式被指定为 XML 或 JSON，服务响应失败可能包含如下表所示的内容。

Element	Child Elements	Value	Description
statusInfo	message	message string	service failure message
	status	success | failure	service status

fmeTransformationResult	fmeServerResponse	FME Server response properties	FME Server response
	fmeEngineResponse	FME Engine response properties	FME Engine response

范例

范例 1

一个数据流服务以 XML 格式响应失败。

请求

```
http://localhost/fmedatastreaming/Samples/austinApartments.fmw?opt_responseformat=xml
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <!-- Service Status Info -->
  <statusInfo>
    <status>failure</status>
    <message> FME Server Transformation Error.</message>
  </statusInfo>
  <fmeTransformationResult>
    <fmeEngineResponse>...</fmeEngineResponse>
    <fmeServerResponse>...</fmeServerResponse>
  </fmeTransformationResult>
</serviceResponse>
```

范例 2

一个数据流服务以 JSON 格式响应失败。

请求

```
http://localhost:/fmedatastreaming/Samples/austinApartments.fmw?opt_responseformat
=json
```

响应

```
{"serviceResponse": {
  "fmeTransformationResult": {
    "fmeEngineResponse": {
      ...
    },
    "fmeServerResponse": {
      ...
    },
    "statusInfo": {
      "message": "FME Server Transformation Error.",
      "status": "failure"
    }
}}
```

数据上传服务

数据上传服务用于集中管理在 FME 转换中使用的文件。 该服务用来上传以及管理文件。 上传文件使用绝对路径，并且绝对路径以一种隐藏了具体敏感的有关机器存储路径的细节形式提供。

在工作区中可以上传文件到这个服务，首先取得上载的文件路径以便用在随后的 FME Server JobSubmitter 转换请求中。Web 用户界面还充分利用数据上传服务来处理涉及源数据的任务。

目前数据上传服务只支持非稳定（易变）存储。 这意味着，上传的文件生存期很短或者说直到闲置一段时间后文件仍是临时的。

此外，上传的文件都存储于由一个唯一的会话 ID 键所指向的专用存储区中。 该会话 ID 键主要是用在所有后续的请求保存会话状态。

数据上传服务只支持上传文件到在服务器上含有存储空间的特定工作区，（用工作区名称来标识这服务器上的存储空间）。 换句话说，如果没有使用工作区名称，便不能使用数据上传服务。

服务请求的组成要素

支持的请求方法

本节描述了与数据上传服务交互可用的各种方法。 发送文件支持的方法是 PUT 和 POST。 HTTP GET 方法支持浏览那些已经上传的文件。

请记住，数据上传服务只支持非稳定（易变）存储，所有的请求必须与 FME Server 工作区相关联。

请求	请求说明	HTTP 方法	Target URI
上传单个文件	您可以使用 HTTP PUT 方法上传一个文件。	PUT	http://host[:port]/fmedataupload/<repository>/<workspace>/<filename>
上传单个或多个文件	你可以通过一个简单的表单提交使用的 HTTP POST 方法上传单个或多个文件	POST	http://host[:port]/fmedataupload/<repository>/<workspace>
浏览上传的文件	通过 GET 请求你可以为一个会话获取所有上传文件的列表，要么使用相同的 URI（统一资源	GET	http://host[:port]/fmedataupload/<repository>/<workspace>/<archive file name>/[<filename or path within archive>]

	标识符）来上传文件原本；要么对存档来说，追加这个存档文件/或该存档的文件路径到文件清单那里。 最后你可以使用可选的部分路径，来显示存档文件中一个子文件夹的内容。要报告的只是存档文件的根路径，部分可选路径可以省略。例如，URI 请求 /fmedataupload/Samples/austinWFS.fmw/myfile.zip/pictures/显示包含名 myfile.zip 存档文件中图片子文件夹下所有文件列表。		

所有服务共同的请求参数

- Workspace Published Parameters
- Transformation Manager Directives
- FME Engine Directives

该服务的具体请求参数

Name	Value	Description

opt_fullpath	true | false Default: false	包含当该服务的响应启用时,FME Engine 可以使用的每个文件的物理或绝对路径。
opt_extractarchive 此参数在 HTTP 的 GET 中不可用，因为提取档案是一个操作，而不是一个状态标识	true | false Default: false	当启用该服务，提取上传 zip 文件并返回 zip 文件中的内容。在这种情况下，文件名是相对于 zip 文件包含路径的逻辑路径。例如,一个文件名为 roads.shp 的 SHAPE 文件在 roads.zip 中，文件返回为 roads.zip / roads.shp。
opt_responseformat	xml | json Default: xml	定义响应的语言。 该文本必须全部小写。
opt_pathlevel	-1 | 0 | n Default: 1	该参数指明显示文件和文件夹信息需要有多少递归层次。 1 表示 1 个完整的递归。 0 表示不返回值任何信息。 任何其他正数表示的递归级别数。 例如，值 1 只请求指定文件路径下的文件，即所要求的路径，而不包含后代子女（即不包含其子文件夹下的文件）。

Preserving Session State 保持会话状态

Java EE 让您明确指定一个请求的 URL 会话 ID。当 Cookie 无法正常工作或不可用时， 该参考能够为会话确保正确的会话状态。 只需简单地追加以下参数作为请求的 URI 路径的一部分：

```
;jsessionid=<session id>
```

范例

下面的例子说明如何使用 jsessionid URI 的实体。 例子中的 URI 太长，会换到下一行并使用一个省略号作为占位符，而占位符是 URI 中微不足道的部分（占位符不影响 URI）。

Basic

```
/fmedataupload/Samples/austinWFS.fmw;jsessionid=6301E253105CD1CE695A7730CD5905B0
```

With Query String

```
/fmedataupload/.../austinWFS.fmw;jsessionid=6301E253105CD1CE695A7730CD5905B0?opt_responseformat=xml
```

服务响应的组成要素

Element	Child Elements	Value	Description
statusInfo	status	success | failure	服务状态
file, folder, archive	name	string	上传的文件名
	size	long integer	上传文件的字节大小，该大小并不适用于文件夹和其下的子文件夹。
	path	string	该文件的绝对或物理路径（用于

			FME Engine)
session	none	string	会话标识符总是用在每个数据上传响应中，作为 HTTP cookie 响应，以及作为主体响应。

Note: There will always be a folder node with an empty name attribute. This folder node is the placeholder for the directory that contains the uploaded files for the scope defined by the current user session, workspace name and repository name. This placeholder can be used later when requesting full paths to retrieve the full path to the parent directory for all uploaded files in your scope.

注：通常总会有文件夹节点的名称属性为空。此文件夹节点是包含当前用户会话、工作区名称和存储名称定义的已上传文件范围目录的占位符。当请求完整路径来为您所有已上传文件的范围内检索父目录的完整路径时，可以使用该占位符。

如果 opt_extractarchive 设置为 false 时，结果会是下列情况：

- 如果被请求的路径检查 zip 或存档文件的内容，则出现一个错误的响应结果。
- 如果请求的路径是有效的，但 opt_pathlevel 是 2 或更高，则 path level 为 1 是不言而喻的。

范例

范例 1

查询示例/ austinWFS.fmw 下所有已上传文件，但不提取任何档案或检查他们的下一级子目录。

请求

```
http://localhost/fmedataupload/Samples/austinWFS.fmw?opt_extractarchive=false
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
    <status>success</status>
  </statusInfo>
  <session>2748D67C99FC2EDBA2160B0215339226</session>
  <files>
    <path />
    <folder>
      <name />
    </folder>
    <archive>
      <size>3826063</size>
      <name>bcalberta.zip</name>
    </archive>
    <archive>
      <size>17337905</size>
      <name>Desktop2.zip</name>
    </archive>
    <archive>
      <size>294731</size>
      <name>parcel_M26.zip</name>
    </archive>
    <archive>
      <size>1374228</size>
      <name>pipes.zip</name>
    </archive>
    <file>
      <size>21684</size>
      <name>csv2shape.fmw</name>
    </file>
    <file>
```

```
    <size>53</size>
    <name>SDS.csv</name>
  </file>
  <file>
    <size>231279</size>
    <name>helmethair.png</name>
  </file>
</files>
</serviceResponse>
```

范例 2

查询示例/ austinWFS.fmw 下所有已上传文件，并且提取当前文件夹下的任何档案。

请求

```
http://localhost/fmedataupload/Samples/austinWFS.fmw?opt_extractarchive=true
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
    <status>success</status>
  </statusInfo>
  <session>2748D67C99FC2EDBA2160B0215339226</session>
  <files>
    <path />
    <folder>
      <name />
    </folder>
    <file>
      <size>21684</size>
      <name>csv2shape.fmw</name>
    </file>
    <file>
      <size>53</size>
```

```
    <name>SDS.csv</name>
  </file>
  <file>
    <size>231279</size>
    <name>helmethair.png</name>
  </file>
  <archive>
    <name>bcalberta.zip</name>
  </archive>
  <archive>
    <archive>
      <name>Desktop2.zip</name>
    </archive>
    <archive>
      <name>parcel_M26.zip</name>
    </archive>
    <archive>
      <name>pipes.zip</name>
    </archive>
  </files>
</serviceResponse>
```

范例 3

检索示例/ austinWFS.fmw 下所有已上传文件的名称以及完全路径，包括三级子目录深度内的任何档案。

请求

```
http://localhost/fmedataupload/Samples/austinWFS.fmw?opt_extractarchive=true&opt_path
level=3&opt_fullpath=true
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
```

```
  <status>success</status>
</statusInfo>
<session>2748D67C99FC2EDBA2160B0215339226</session>
<files>
  <path />
  <folder>
    <name />
    <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
97885FBD23A32E245/</path>
  </folder>
  <file>
    <size>70013</size>
    <name>bcalberta.zip/bcalberta/AEROFACP.ffs</name>
    <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
97885FBD23A32E245
/__fmearch_name_bcalberta.zip/bcalberta/AEROFACP.ffs</path>
  </file>
  <file>
    <size>1010226</size>
    <name>bcalberta.zip/bcalberta/COASTL.ffs</name>
    <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
97885FBD23A32E245
/__fmearch_name_bcalberta.zip/bcalberta/COASTL.ffs</path>
  </file>
  <file>
    <size>578807</size>
    <name>bcalberta.zip/bcalberta/INWATERA.fsi</name>
    <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
97885FBD23A32E245
/__fmearch_name_bcalberta.zip/bcalberta/INWATERA.fsi</path>
  </file>
  <archive>
    <name>bcalberta.zip</name>
    <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
97885FBD23A32E245
/__fmearch_name_bcalberta.zip</path>
  </archive>
  <folder>
```

```
      <name>bcalberta.zip/bcalberta</name>
      <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
97885FBD23A32E245
/__fmearch_name_bcalberta.zip/bcalberta</path>
    </folder>
  </files>
</serviceResponse>
```

范例 4

查询示例/ austinWFS.fmw 下标题名为 bcalberta.zip、目录深度为三级的所有已上传文件，包括三级子目录深度内以及所有嵌套的档案文件，同时提取所有嵌套的档案文件。

请求

```
http://localhost/fmedataupload/Samples/austinWFS.fmw/bcalberta.zip?opt_extractarch
ive=true&opt_pathlevel=3
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
    <status>success</status>
  </statusInfo>
  <session>2748D67C99FC2EDBA2160B0215339226</session>
  <files>
    <path>/bcalberta.zip</path>
    <folder>
      <name />
    </folder>
    <file>
      <size>70013</size>
      <name>bcalberta/AEROFACP.ffs</name>
    </file>
    <file>
```

```
    <size>1010226</size>
    <name>bcalberta/COASTL.ffs</name>
  </file>
  <file>
    <size>578807</size>
    <name>bcalberta/INWATERA.fsi</name>
  </file>
  <folder>
    <name>bcalberta</name>
  </folder>
 </files>
      </serviceResponse>
```

范例 5

检索示例/ austinWFS.fmw 下所有上传文件的名称和完整路径包括之前三级子目录深度内提取的文件和文件夹。 它不涉及嵌套包含目录范围内的任何档案提取，所以说， 三级深度只有当一个请求是以前提取过任何相关的档案时才变得有意义。

请求

```
http://localhost/fmedataupload/Samples/austinWFS.fmw?opt_extractarchive=false&opt_pathlevel=3&opt_fullpath=true
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
    <status>success</status>
  </statusInfo>
  <session>2748D67C99FC2EDBA2160B0215339226</session>
  <files>
    <path />
    <folder>
      <name />
      <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
```

```
97885FBD23A32E245/</path>
    </folder>
    <archive>
      <name>bcalberta.zip</name>
      <path>$(FME_DATA_REPOSITORY)/Samples/austin.fmw/AB4F3D23F
97885FBD23A32E245/bcalberta.zip</path>
    </archive>
  </files>
</serviceResponse>
```

范例 6

检索示例/ austinWFS.fmw 下所有上传文件的名称包括之前三级子目录深度内提取的文件和文件夹。 它不涉及嵌套包含目录范围内的任何档案提取，所以说， 三级深度只有当一个请求是以前提取过任何相关的档案时才变得有意义。

请求

```
http://localhost/fmedataupload/Samples/austinWFS.fmw?opt_extractarchive=false&opt_
pathlevel=3
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
    <status>success</status>
  </statusInfo>
  <session>2748D67C99FC2EDBA2160B0215339226</session>
  <files>
    <path />
    <folder>
      <name />
    </folder>
    <archive>
      <size>3826063</size>
      <name>bcalberta.zip</name>
```

```
    </archive>
    <archive>
      <size>17337905</size>
      <name>Desktop2.zip</name>
    </archive>
    <archive>
      <size>294731</size>
      <name>parcel_M26.zip</name>
    </archive>
    <archive>
      <size>1374228</size>
      <name>pipes.zip</name>
    </archive>
    <file>
      <size>21684</size>
      <name>csv2shape.fmw</name>
    </file>
    <file>
      <size>53</size>
      <name>SDS.csv</name>
    </file>
    <file>
      <size>231279</size>
      <name>helmethair.png</name>
    </file>
  </files>
</serviceResponse>
```

范例 7

试图查询示例/ austinWFS.fmw 下 bcalberta.zip 文件的内容，但失败。 失败是基于这样的推定，bcalberta.zip 归档文件以前未提取过，而且该请求的可选参数未指定要提取任何档案。

请求

```
http://localhost/fmedataupload/Samples/austinWFS.fmw/bcalberta.zip?opt_extractarchive=false
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
    <status>failure</status>
  </statusInfo>
  <session>2748D67C99FC2EDBA2160B0215339226</session>
</serviceResponse>
```

任务提交服务

任务提交服务接受并运行工作区中的任务请求。

服务请求的组成要素

该服务请求的 URL 地址

```
http://host[:port]/fmejobsubmitter/<repository>/<workspace>?{name[=value]&}
```

所有服务共同的请求参数

- Workspace Published Parameters
- Transformation Manager Directives
- FME Engine Directives

该服务的具体请求参数

下表描述了该服务的具体可选参数项。

Name	Value	Description
opt_responseformat	xml | json | html	该服务的响应格式
opt_servicemode	sync | async	在同步和异步服务模式之间切换
opt_requesteremail	comma-separated, e-mail addresses	发送通知电子邮件的地址

服务响应的组成要素

如果响应格式被指定为 XML 或 JSON，服务响应可能包含如下表所示的内容：：

Element	Child Elements	Value	描述
Status Info	message	message string	服务失败消息
	status	success | failure	服务状态
	mode	sync | async	服务模式

Fme Trans formati on Result	fmeServerResponse	FME Server response properti es	FME Server 响应
	fmeEngineResponse	FME Engine response properti es	FME Engine 响应
email	none	Email addresse s	请求者的电 子邮件地址
jobID	none	job ID	当前任务标 识符

范例

范例 1

展示了一个指定 XML 作为响应格式的任务提交服务请求。

请求

```
http://localhost/fmedatadownload/Samples/austinDownload.fmw?THEMES=railroad&opt_responseformat=xml
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
```

```
<!-- Service Status Info -->
<statusInfo>
  <status>success</status>
  <mode>sync</mode>
</statusInfo>

<!-- Service Return -->
<fmeTransformationResult>
    <fmeEngineResponse>...</fmeEngineResponse>
    <fmeServerResponse>...</fmeServerResponse>
  </fmeTransformationResult>
  <jobID>53</jobID>
</serviceResponse>
```

范例 2

展示了一个参数带有请求者的 e - mail 地址，服务为异步模式，以及指定 JSON 作为响应格式的任务提交服务请求。

请求

```
http://localhost/fmejobsubmitter/Samples/austinDownload.fmw?opt_servicemode=async&opt_requesteremail=joe@safe.com&opt_responseformat=xml
```

响应

```
{"serviceResponse": {
   "email": "someone@safe.com",
   "jobID": 15,
   "statusInfo": {
      "mode": "async",
      "status": "success"
   }
}}
```

KML 网络链接服务

KML 网络链接服务接受请求并创建一个 KML 网络链接文件。 它返回一个包含该 KML 网络链接文件作为数据流的 KMZ 文件。

服务请求的组成要素

该服务请求的 URL 地址

```
http://host[:port]/fmekmllink/<repository>/<workspace>?{name[=value]&}
```

所有服务共同的请求参数

- Workspace Published Parameters
- Transformation Manager Directives
- FME Engine Directives

该服务的具体请求参数

下表描述了该服务的具体可选参数项。

Name	Value	Description
opt_responseformat	xml | json | html	该服务的响应格式

当该服务执行成功时，网络链接服务服务以数据流形式返回 KMZ 结果数据集从而 opt_responseformat 参数将被忽略； 当

服务执行失败时，此参数指定将要返回的响应格式：XML，JSON 或 HTML。

工作区发布参数

该网络链接服务适用于任何写入 KML 和 KMZ 文件的工作空间。如下表所示的发布参数，应该在工作区中存在。

Name	Value	Description
bboxEast	coordinate value	Bounding box east
bboxWest	coordinate value	Bounding box west
bboxNorth	coordinate value	Bounding box north
bboxSouth	coordinate value	Bounding box south

服务响应的组成要素

如果 KML 网络链接服务成功处理了请求，它返回一个 KMZ 格式封装的 KML 文件作为数据流给客户端。该 KML 文件包含一个网络链接元素，其中的 Link/href 设置为一个 URL。 该 URL 调用由库和工作区名称指定的 FME Server 数据流服务。 对于完整的 KML 格式参考，请参阅：

http://code.google.com/apis/kml/documentation/

如果响应格式被指定为 XML 或 JSON，服务失败的响应可能包含如下表所示的内容。

Element	Child Elements	Value	Description
statusInfo	message	message string	服务失败消息
	status	success | failure	服务状态

范例

范例 1

以 KML 格式成功响应一个 KML 网络链接请求。

请求

```
http://localhost/fmekmllink/Samples/austinApartments.fmw
```

响应

```
<?xml version="1.0" encoding="UTF-8" ?>
  <kml xmlns="http://earth.google.com/kml/2.1">
    <NetworkLink>
    <name>KML Link Translation: austinApartments.fmw</name>
    <visibility>0</visibility>
    <description>A KML Link document processed by FME
Server</description>
      <Link>
      <viewRefreshMode>onRegion</viewRefreshMode>
      <viewRefreshTime>7</viewRefreshTime>
```

```
      <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxN
orth]</viewFormat>
      <href>http://SHOCKWAVE:8080/fmedatastreaming/Samples/aust
inApartments.fmw</href>
      <refreshMode>onInterval</refreshMode>
      <refreshInterval>60</refreshInterval>
    </Link>
  </NetworkLink>
</kml>
```

范例 2

以 XML 格式响应一个失败的 KML 网络链接请求。

请求

```
http://localhost/fmekmllink/Samples/austinApartments2.fmw?opt_responseformat=xml
```

响应

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<serviceResponse>
  <statusInfo>
    <message>Workspace 'austinApartments2.fmw' does not
exist.</message>
    <status>failure</status>
  </statusInfo>
</serviceResponse>
```

范例 3

以 JSON 格式响应一个失败的 KML 网络链接请求。

请求

```
http://localhost/fmekmllink/Samples/austinApartments2.fmw?opt_responseformat=json
```

响应

```
{"serviceResponse": {"statusInfo": {
   "message": "Workspace 'austinApartments2.fmw' does not exist.",
   "status": "failure"
}}}
```

OGC Web 要素服务

开放地理空间联盟（OGC）的 Web 要素服务（WFS）以函数形式提供 WFS 功能。

支持的版本

FME Server WFS service 支持 OGC WFS versions 1.1.0 and 1.0.0.

服务的功能

FME Server WFS service 支持如下操作:

- GetCapabilities,
- DescribeFeatureType, and
- GetFeature.

服务请求的组成要素

FME Server WFS 支持的请求格式和参数与 OGC WFS 标准完全兼容。

该服务请求的 URL 地址

```
http://host[:port]/fmeogc/<repository>/<workspace>?{name[=value]&}
```

该服务请求的编码

下表显示了操作请求编码。请记住 KVP 指的是 keyword-value pair（关键字-值对）。

Operation	Request Encoding
GetCapabilities	XML & KVP
DescribeFeatureType	XML & KVP
GetFeature	XML & KVP

下表显示了请求编码和 HTTP 方法。

Request Encoding	HTTP GET method	HTTP POST method
XML encoded requests	not applicable	text/xml
KVP	query	application/x-www-form-urlenco

encoded request s	paramete rs encoding	ded

范例

范例 1

GetCapabilities 请求示例

```
http://localhost/fmeogc/Samples/austinWFS.fmw?SERVICE=WFS&REQUEST=GetCapabilities&VERSION=1.1.0
```

范例 2

DescribeFeatureType 请求示例

```
http://localhost/fmeogc/Samples/austinWFS.fmw?SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&TYPENAME=fme%3Aairports%2Cfme%3Acenart
```

范例 3

GetFeature 请求示例

```
http://localhost/fmeogc/Samples/austinWFS.fmw?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=fme%3Aairports%2Cfme%3Acenart
```

OGC Web 地图服务

OGC Web Map Service (WMS)以函数形式提供 WMS 功能。

支持的版本

支持的请求格式和参数与 OGC WMS 标准 1.1.1、1.3.0.完全兼容。

服务的功能

FME Server WMS 支持如下操作:

- GetCapabilities and
- GetMap.

服务请求的组成要素

FME Server WMS 支持的请求格式和参数与 OGC WMS 标准完全兼容。

该服务请求的 URL 地址

```
http://host[:port]/fmeogc/<repository>/<workspace>?{name[=value]&}
```

该服务请求的编码

下表显示了操作请求编码。

Operation	Request Encoding

GetCapabilities	XML & KVP
GetMap	XML & KVP

响应格式

下表显示了 FME Server WMS 支持的格式。

Format Name	Full Name	MIME Type
GIF	Graphics Interchange Format	image/gif
PNG	Portable Network Graphics	image/png
JPEG	Joint Photographics Expert Group	Image/jpeg
SVG	Scalable Vector Graphics	image/svg+xml

范例

范例 1

GetCapabilities Request 示例

```
http://localhost/fmeogc/Samples/austinWMS.fmw?SERVICE=WMS&REQUEST=GetCapabilities&VERSION=1.1.1
```

范例 2

GetMap Request 示例

http://localhost/fmeogc/Samples/austinWMS.fmw?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&LAYERS=airports,cenart&STYLES=&FORMAT=image/png&SRS=EPSG:4326&BBOX=98.7,29,96.5,31.5&WIDTH=560&HEIGHT=300

COM.safe.webservices.soap.servlet

Interface IFMEServerSoapService

`public interface` IFMEServerSoapService

This interface defines the methods available on FME Server SOAP API. These methods are described by the Web Services Description Language (WSDL), which you can access using the following URL through FME Server SOAP service:
http://host[:port]/fmesoap/IFMEServerSoapService?wsdl

Response Format

If a service call succeeds, the return message is in the form that follows:
SUCCESS<SOAP message delimiter><message body>

If a service call fails, the return message is in the form that follows:
ERROR<SOAP message delimiter><message body>

For example, a successful **getRepositories**() call may return the following response in the SOAP message body:
SUCCESS[delim]<?xml version="1.0" encoding="UTF-8"?>
<REP_GET_REPOSITORIES><REPOSITORY name="Samples" description=""/></REP_GET_REPOSITORIES>

SOAP Message Delimiter

The delimiter in the current FME Server SOAP API is **[delim]**, which is used for:
- separating the name and value pairs in all directives, when the directives

are input parameters of methods
- separating the message header and body of the SOAP response message

---

Method Summary	
java.lang.String	**abortJob**(java.lang.String jobID) Cancels the given job ID.
java.lang.String	**addItem**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String crcEnabled, javax.activation.DataHandler fileAttachment) Adds (uploads) a new item to the repository.
java.lang.String	**addRepository**(java.lang.String repositoryName, java.lang.String description) Adds a repository to FME Server.
java.lang.String	**addResource**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String resourceName, java.lang.String crcEnabled, javax.activation.DataHandler fileAttachment) Adds (uploads) a new resource to an item that's inside the specified repository.
java.lang.String	**addService**(java.lang.String serviceName, java.lang.String displayName, java.lang.String description, java.lang.String urlPattern, java.lang.String isEnabled, java.lang.String isWorkspaceRegAllowed) Creates a new service on FME Server.
java.lang.String	**deregisterWorkspaceService**(java.lang.String repositoryName, java.lang.String workspaceName, java.lang.String serviceName) Removes the registration of a single service with a single workspace.
java.lang.String	**getCustomFormat**(java.lang.String repositoryName, java.lang.String formatName) Gets a custom format in a specified repository.
java.lang.String	**getCustomFormats**(java.lang.String repositoryName, java.lang.String directives) Gets all custom formats inside a specified repository.

java.lang.String	**getCustomTransformer**(java.lang.String repositoryName, java.lang.String transformerName) Gets a custom transformer in a specified repository.
java.lang.String	**getCustomTransformers**(java.lang.String repositoryName, java.lang.String directives) Gets all custom transformers in a specified repository.
java.lang.String	**getGlobalAllItems**(java.lang.String enableFilter) Gets all item names from all repositories.
javax.activation.DataHandler	**getItem**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String crcEnabled) Gets (download) an item from the repository.
java.lang.String	**getJobResult**(java.lang.String jobID) Gets the result string produced by FME Server after a job has completed successfully or unsuccessfully.
java.lang.String	**getRepositories**(java.lang.String directives) Returns the list of all repositories available on FME Server.
java.lang.String	**getRepository**(java.lang.String repositoryName) Returns the information of a given repository on FME Server.
javax.activation.DataHandler	**getResource**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String resourceName, java.lang.String crcEnabled) Gets (downloads) a resource that belongs to an item in the specified repository.
java.lang.String	**getResourcesView**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String directives) Gets all the summaries of all resources for the an item in the specified repository.
java.lang.String	**getResourceView**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String resourceName) Gets the summary of a resource belonging to an item in the specified repository.
java.lang.String	**getService**(java.lang.String serviceName) Gets the information about a given service on FME

	Server. This method returns one of these services.
java.lang.String	**getServices**(java.lang.String directives) Gets the list of all services available from FME Server.
java.lang.String	**getWorkspace**(java.lang.String repositoryName, java.lang.String workspaceName) Gets a workspace object (serialized as XML) in a specified repository.
java.lang.String	**getWorkspaces**(java.lang.String repositoryName, java.lang.String directives) Gets the workspace objects (serialized as XML) of every workspace in a specified repository.
java.lang.String	**getWorkspaceServices**(java.lang.String repositoryName, java.lang.String directives) Returns the summary of workspace registrations in a specified repository.
java.lang.String	**getWorkspaceSummaries**(java.lang.String repositoryName, java.lang.String directives) Returns the summaries for all the workspaces available inside the specified repository.
java.lang.String	**itemExists**(java.lang.String repositoryName, java.lang.String itemName) Determines if the specified item exists.
java.lang.String	**registerWorkspaceService**(java.lang.String repositoryName, java.lang.String workspaceName, java.lang.String serviceName) Register a specified workspace with a specified service.
java.lang.String	**removeItem**(java.lang.String repositoryName, java.lang.String itemName) Removes an item from the specified repository.
java.lang.String	**removeRepository**(java.lang.String repositoryName) Removes the specified repository from FME Server. **Note:** Removing a repository also removes any objects (workspaces, custom formats, custom transformers, data, etc) stored inside that repository.
java.lang.String	**removeResource**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String resourceName) Removes a resource from an item in the specified

	repository.
java.lang.String	**removeService**(java.lang.String serviceName) Removes the specified service from FME Server. **Note:** Removing a service also causes any workspaces registered with it to become unregistered.
java.lang.String	**repositoryExists**(java.lang.String repositoryName) Determines if the specified repository exists.
java.lang.String	**resourceExists**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String resourceName) Determines if the specified resource exists.
java.lang.String	**serviceExists**(java.lang.String serviceName) Determines if the specified service exists.
java.lang.String	**sessionInit**(java.lang.String connectionInfo, java.lang.String credentialType) Attempts to connect to FME Server using the given credentials and credential type.
java.lang.String	**submitJob**(java.lang.String subsection, java.lang.String workspacePath, java.lang.String publishedParameters, java.lang.String fmeEngineDirectives, java.lang.String transformationManagerDirectives) Submits a workspace transformation job to FME Server.
java.lang.String	**transactJob**(java.lang.String subsection, java.lang.String workspacePath, java.lang.String publishedParameters, java.lang.String fmeEngineDirectives, java.lang.String transformationManagerDirectives) Invokes a workspace transformation job.
java.lang.String	**updateItem**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String crcEnabled, javax.activation.DataHandler fileAttachment) Updates an item that already exists on FME Server.
java.lang.String	**updateResource**(java.lang.String repositoryName, java.lang.String itemName, java.lang.String resourceName,

	java.lang.String crcEnabled, javax.activation.DataHandler fileAttachment) Updates an existing resource to an item inside the specified repository.
java.lang.String	**updateService**(java.lang.String serviceName, java.lang.String displayName, java.lang.String description, java.lang.String urlPattern, java.lang.String isEnabled, java.lang.String isWorkspaceRegAllowed) Updates a service on FME Server.
java.lang.String	**workspaceServceExists**(java.lang.String repositoryName, java.lang.String workspaceName, java.lang.String serviceName) Determines if a specified workspace is registered with a specified service.

Method Detail

removeService

java.lang.String **removeService**(java.lang.String serviceName)
throws java.lang.Exception

Removes the specified service from FME Server.
Note: Removing a service also causes any workspaces registered with it to become unregistered.
Parameters:

serviceName - The name of the service you want to remove.

Returns:
'true' if the service was removed successfully, otherwise 'false'.
Throws:
java.lang.Exception

---

serviceExists

java.lang.String **serviceExists**(java.lang.String serviceName)
throws java.lang.Exception

Determines if the specified service exists.

Parameters:

`serviceName` - The name of the service being searched for.

Returns:
'true' if service exists; otherwise 'false' if service does not exist
Throws:
`java.lang.Exception`

---

getServices

`java.lang.String` **`getServices`**`(java.lang.String directives)`
`throws java.lang.Exception`

Gets the list of all services available from FME Server.
Parameters:

`serviceName` - The name of the service being searched for.

Returns:
The response from FME Server to the getServices command.
-A successful result is an XML document containing all the services.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.
Throws:
`java.lang.Exception`

---

getService

`java.lang.String` **`getService`**`(java.lang.String serviceName)`
`throws java.lang.Exception`

Gets the information about a given service on FME Server.
This method returns one of these services.
Parameters:

`serviceName` - The name of the service being searched for.

Returns:
The response from FME Server to the getService command.
-A successful result is an XML serialization of a service object.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.
Throws:
`java.lang.Exception`

---

addService

java.lang.String **addService**(java.lang.String serviceName,
java.lang.String displayName,
java.lang.String description,
java.lang.String urlPattern,
java.lang.String isEnabled,
java.lang.String isWorkspaceRegAllowed)
throws java.lang.Exception

Creates a new service on FME Server.

Parameters:

serviceName - The name of the new service. This is the name that you use to access the service with using the API or FMEServerConsole.

displayName - The display name of the service. This is the name that will appear in the FMEServer Web UI.

urlPattern - The URL pattern of the service. This is the URL that you want to relate workspaces to.

description - A short description of the new service.

isEnabled - A 'true' or 'false' value indicating whether the service is enabled.

allowWorkspaceRegistration - A 'true' or 'false' value indicating whether the service allows workspace registration.

Returns:

The response from FME Server to the addService command.
-A successful result is an XML serialization of the service name.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

updateService

java.lang.String **updateService**(java.lang.String serviceName,
java.lang.String displayName,
java.lang.String description,
java.lang.String urlPattern,

java.lang.String isEnabled,

java.lang.String isWorkspaceRegAllowed)

throws java.lang.Exception

Updates a service on FME Server.

Parameters:

serviceName - The name of the service to update. This is the name that you access the service with via the API or FMEServerConsole.

commonName - The new display or common name of the service. This is the name that will appear in the FMEServer Web UI.

urlPattern - The new URL pattern of the service. This is the URL that you want to relate workspaces to.

description - A new short description of the new service.

isEnabled - A 'true' or 'false' value indicating whether the service is enabled.

allowWorkspaceRegistration - A 'true' or 'false' value indicating whether the service allows workspace registration.

Returns:

The response from FME Server to the updateService command.
-A successful result is an XML serialization of the new service name.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

repositoryExists

java.lang.String **repositoryExists**(java.lang.String repositoryName)

throws java.lang.Exception

Determines if the specified repository exists.

Parameters:

repositoryName - The name of the repository being searched for.

Returns:

The response to the repositoryExists command.
-'True' if the a repository exists with the specified name.
-'False' if the repository does not exist.

-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

getRepositories

java.lang.String **getRepositories**(java.lang.String directives)
throws java.lang.Exception

Returns the list of all repositories available on FME Server.

Parameters:

directives - A list of name/value pairs delimited by the SOAP message delimiter. The available directives are:
- Name: 'enableFilter',
Value: 'All'

Returns:

The response from FME Server to the getRepositories command.
-A successful result is an XML list of repositories.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

getRepository

java.lang.String **getRepository**(java.lang.String repositoryName)
throws java.lang.Exception

Returns the information of a given repository on FME Server.

Parameters:

repositoryName - The name of the repository being searched for.

Returns:

The response from FME Server to the getRepositry command.
-A successful result is an XML serialization of a repository object.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

addRepository

java.lang.String **addRepository**(java.lang.String repositoryName,
java.lang.String description)
throws java.lang.Exception

Adds a repository to FME Server.

Parameters:

repositoryName - The name of the repository to create. This name is used by the API and FME ServerConsole to access the repository. The FME Server Web UI will use this name to display show about the repository and objects inside the repository.

description - A short description of what the repository is to be used for.

Returns:

The response from FME Server to the addRepository command.
-A successful result is the name of the newly created repository. However, more important then the result of this command is that a new repository is created on FME Server that can be accessed by the ServerSoapService#getRepositories() orgetRepository(String) methods.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

removeRepository

java.lang.String **removeRepository**(java.lang.String repositoryName)
throws java.lang.Exception

Removes the specified repository from FME Server.

Note: Removing a repository also removes any objects (workspaces, custom formats, custom transformers, data, etc) stored inside that repository. Be careful to back up anything you need before removing it.

Parameters:

repositoryName - The name of the repository you want to remove.

Returns:

The response from FME Server to the removeRepository command.

-"true" if the repository was successfully removed.
-"false" if the repository could not be found.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

itemExists

```
java.lang.String itemExists(java.lang.String repositoryName,
                            java.lang.String itemName)
                            throws java.lang.Exception
```

Determines if the specified item exists. Items are anything that can be stored in a repository (workspaces, custom formats, custom transformers, published data).

Parameters:

repositoryName - The name of the repository to search for the item.

itemName - The name of the item to search for.

Returns:

'true' if the item exists; otherwise 'false' if the item does not exist

Throws:

java.lang.Exception

---

removeItem

```
java.lang.String removeItem(java.lang.String repositoryName,
                            java.lang.String itemName)
                            throws java.lang.Exception
```

Removes an item from the specified repository.

Parameters:

repositoryName - The name of the repository for the item.

itemName - The name of the item.

Returns:

'true' if the item was successfully removed, otherwise 'false'.

Throws:

java.lang.Exception

---

getItem

```
javax.activation.DataHandler getItem(java.lang.String repositoryName,
                                     java.lang.String itemName,
                                     java.lang.String crcEnabled)
                                     throws java.lang.Exception
```

Gets (download) an item from the repository.

Parameters:

`repositoryName` - The name of the repository holding the object.

`itemName` - The name of the item inside the repository.

`itemFileCRCCheckingEnabled` - A 'true' or 'false' value indicating whether the file is enabled for CRC checking. The default value is 'false'.

Returns:

The response from FME Server to the getItem command.

-A successful result comes with the item file as an MTOM attachment.

-An error message is returned if the item cannot be found or if a fatal exception is thrown when attempting to complete this request.

Throws:

`java.lang.Exception`

---

addItem

```
java.lang.String addItem(java.lang.String repositoryName,
                         java.lang.String itemName,
                         java.lang.String crcEnabled,
                         javax.activation.DataHandler fileAttachment)
                         throws java.lang.Exception
```

Adds (uploads) a new item to the repository.

Parameters:

`repositoryName` - The name of the repository to which you want to add the item.

`itemName` - The name of the item you want to add.

`itemFileCRCCheckingEnabled` - A 'true' or 'false' value indicating whether the file is enabled for CRC checking. The default value is

'false'.

`fileAttachment` - An MTOM attachment containing the item file.

Returns:
The response from FME Server to the addItem command.
-A successful result indicates that the item was added to the repository.
Throws:
`java.lang.Exception`

---

updateItem

```
java.lang.String updateItem(java.lang.String repositoryName,
                            java.lang.String itemName,
                            java.lang.String crcEnabled,

javax.activation.DataHandler fileAttachment)
                            throws java.lang.Exception
```

Updates an item that already exists on FME Server.
Note: Any changes made to an item will completely overwrite the existing item. If you want to save the old item please remember to back it up before calling this method.
Parameters:

`repositoryName` - The name of the repository to which to which you want to update the item.

`itemName` - The name of the item to which you want to update.

`itemFileCRCCheckingEnabled` - A 'true' or 'false' value indicating if the file is enabled for CRC checking. The default value is 'false'.

`fileAttachment` - An MTOM attachment containing the item file.

Returns:
The response from FME Server to the addItem command.
- A successful result indicates that the item was updated in the specified repository.
Throws:
`java.lang.Exception`

---

getWorkspaceSummaries

java.lang.String
getWorkspaceSummaries(java.lang.String repositoryName,
java.lang.String directives)
throws java.lang.Exception

Returns the summaries for all the workspaces available inside the specified repository.

Parameters:

repositoryName - The name of the repository to search.

directives - A list of name/value pairs delimited by the SOAP message delimiter. The available directives are:
- Name: 'enableFilter',
Value: 'All'

Returns:

The response from FME Server to the getWorkspaceSummaries command.
-A successful result is an XML serialized list of workspace summaries.
-An error message is returned if the item cannot be found or if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

getWorkspaces

java.lang.String **getWorkspaces**(java.lang.String repositoryName,
java.lang.String directives)
throws java.lang.Exception

Gets the workspace objects (serialized as XML) of every workspace in a specified repository.

Parameters:

repositoryName - The name of the repository to search for workspaces

directives - A list of name/value pairs delimited by the SOAP message delimiter. The available directives are:
- Name: 'enableFilter',
Value: 'All'

Returns:

The response from FME Server to the getWorkspaces command.
-A successful result is an XML serialized list of workspace summaries.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

getWorkspace

java.lang.String **getWorkspace**(java.lang.String repositoryName,
java.lang.String workspaceName)
throws java.lang.Exception

Gets a workspace object (serialized as XML) in a specified repository.

Parameters:

repositoryName - The name of the repository

workspaceName - The name of the workspace

Returns:

The response from FME Server to the getWorkspaces command.
-A successful result is an XML serialized list of workspace summaries.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

workspaceServceExists

java.lang.String
workspaceServceExists(java.lang.String repositoryName,
java.lang.String workspaceName,
java.lang.String serviceName)
throws java.lang.Exception

Determines if a specified workspace is registered with a specified service.

Parameters:

serviceName - The service we are looking to see if the workspace is registered with.

repositoryName - The name of the repository where the workspace resides.

workspaceName - The name of the workspace being tested.

Returns:

The response from FME Server to the workspaceServiceExists command

-"true" if the specified workspace is registered to the specified service.

-"false" otherwise

-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

getWorkspaceServices

java.lang.String
getWorkspaceServices(java.lang.String repositoryName,
java.lang.String directives)
throws java.lang.Exception

Returns the summary of workspace registrations in a specified repository.

Parameters:

repositoryName - The name of the repository to check.

directives - A list of name/value pairs delimited by the SOAP message delimiter. The available directives are:
- Name: 'enableFilter',
Value: 'All'

Returns:

The response from FME Server to the getWorkspaceServices command.

-A successful result will be an XML list of workspace services all of which registered to workspaces inside the specified repository.

-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

registerWorkspaceService

java.lang.String
registerWorkspaceService(java.lang.String repositoryName,

java.lang.String workspaceName,

java.lang.String serviceName)

throws java.lang.Exception

Register a specified workspace with a specified service.

Parameters:

repositoryName - The name of the repository in which the workspace resides in.

workspaceName - The name of the specified workspace to register.

serviceName - The name of the service we want to register the URL against.

Returns:

The response from FME Server to the registerWorkspace command.
-"true" if the workspace was successfully registered to the service. However, more importantly then the result is that the FME Server Web UI has the option to run the specified workspace on the services URL.
-"false" if for some reason the workspace could not be registered.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

deregisterWorkspaceService

java.lang.String
deregisterWorkspaceService(java.lang.String repositoryName,

java.lang.String workspaceName,

java.lang.String serviceName)

throws
java.lang.Exception

Removes the registration of a single service with a single workspace.

Parameters:

`repositoryName` - The name of the repository in which the workspace resides.

`workspaceName` - The name of the workspace.

`serviceName` - The name of the service we want to deregister the workspace from.

Returns:

The response from FME Server to the deregisterWorkspace command.
-"true" if the workspace was successfully deregistered to the service. However, more importantly then the result is that the FME Server Web UI has the option to run the specified workspace on the services URL.
-"false" if for some reason the workspace could not be deregistered.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

getCustomFormats

java.lang.String **getCustomFormats**(java.lang.String repositoryName,
java.lang.String directives)
throws java.lang.Exception

Gets all custom formats inside a specified repository.

Parameters:

`repositoryName` - The name of the repository to search for custom formats.

`directives` - A list of name/value pairs delimited by the SOAP message delimiter. The available directives are:
- Name: 'enableFilter',
Value: 'All'

Returns:

The response from FME Server to the getCustomFormats command.
-A successful result is an XML serialized list of custom formats

available inside the repository.
-If no custom formats exist inside the repository, an empty list will be returned.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.
Throws:
java.lang.Exception

---

getCustomFormat

java.lang.String **getCustomFormat**(java.lang.String repositoryName,
java.lang.String formatName)
throws java.lang.Exception

Gets a custom format in a specified repository.
Parameters:

repositoryName - The name of the repository where the format resides.

formatName - The name of the custom format.

Returns:
The response from FME Server to the getCustomFormat command.
-A successful result is an XML serialized representation of a customFomat.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.
Throws:
java.lang.Exception

---

getCustomTransformers

java.lang.String
getCustomTransformers(java.lang.String repositoryName,
java.lang.String directives)
throws java.lang.Exception

Gets all custom transformers in a specified repository.
Parameters:

repositoryName - The name of the repository to be searched.

directives - A list of name/value pairs delimited by the SOAP

message delimiter. The available directives are:
- Name: 'enableFilter',
Value: 'All'

Returns:

The response from FME Server to the getCustomTransformers command.
-A successful result is an XML serialized list of custom transformers.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

getCustomTransformer

java.lang.String
getCustomTransformer(java.lang.String repositoryName,

java.lang.String transformerName)
throws java.lang.Exception

Gets a custom transformer in a specified repository.

Parameters:

repositoryName - The name of the repository to be searched.

transformerName - The name of the custom transformer.

Returns:

The response from FME Server to the getCustomTransformers command.
-A successful result is an XML serialized custom transformer.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

transactJob

java.lang.String **transactJob**(java.lang.String subsection,
java.lang.String workspacePath,
java.lang.String publishedParameters,
java.lang.String fmeEngineDirectives,

java.lang.String transformationManagerDirectives)
throws java.lang.Exception

Invokes a workspace transformation job.

Parameters:

repositoryName - The repository where the workspace is located.

workspacePath - The name of the workspace used in the translation.

subsection - The subsection name for this job.

publishedParameters - The workspace's published parameters encoded in a list of name/value pairs delimited by the SOAP message delimiter.

fmeEngineDirectives - The FME Engine directives encoded in a list of name/value pairs delimited by the SOAP message delimiter.

transformationManagerDirectives - The Transformation Manager directives encoded in a list of name/value pairs delimited by the SOAP message delimiter.

Returns:

The transformation result string generated by FME Server.

Throws:

java.lang.Exception

---

submitJob

java.lang.String **submitJob**(java.lang.String subsection,
java.lang.String workspacePath,
java.lang.String publishedParameters,
java.lang.String fmeEngineDirectives,
java.lang.String transformationManagerDirectives)
throws java.lang.Exception

Submits a workspace transformation job to FME Server.

Parameters:

repositoryName - The repository where the workspace is located.

workspacePath - The name of the workspace used in the translation.

`subsection` - The subsection name for this job.

`publishedParameters` - The workspace's published parameters encoded in a list of name/value pairs delimited by the SOAP message delimiter.

`fmeEngineDirectives` - The FME Engine directives encoded in a list of name/value pairs delimited by the SOAP message delimiter.

`transformationManagerDirectives` - The Transformation Manager directives encoded in a list of name/value pairs delimited by the SOAP message delimiter.

Returns:

The unique ID for the job.

Throws:

`java.lang.Exception`

---

getJobResult

`java.lang.String` **`getJobResult`**`(java.lang.String jobID)`
`throws java.lang.Exception`

Gets the result string produced by FME Server after a job has completed successfully or unsuccessfully.

Note: This method will block until the job is complete if it is not already complete.

Parameters:

`jobID` - The number used to uniquely identify the job whose result you want to get.

Returns:

The result string as specified by the subsection used to submit this job.

Throws:

`java.lang.Exception`

---

abortJob

`java.lang.String` **`abortJob`**`(java.lang.String jobID)`
`throws java.lang.Exception`

Cancels the given job ID.

Parameters:

`jobID` - The unique ID used to identify the job.

Returns:
-"true" if the job was successfully canceled.
-"false" if the job was not canceled.
-An error message is returned if a fatal exception is thrown when attempting to complete this request.
Throws:
java.lang.Exception

---

addResource

```
java.lang.String addResource(java.lang.String repositoryName,
                             java.lang.String itemName,
                             java.lang.String resourceName,
                             java.lang.String crcEnabled,

javax.activation.DataHandler fileAttachment)
                             throws java.lang.Exception
```

Adds (uploads) a new resource to an item that's inside the specified repository.
Parameters:

`repositoryName` - The name of the repository where the item resides.

`itemName` - The name of the item to add the resource to.

`resourceName` - The name of the new resource.

`resourceFileCRCCheckingEnabled` - A 'true' or 'false' value indicating if the resource file is CRC checking enabled. The default value is 'false'.

`fileAttachment` - An MTOM attachment containing the resource file.
Returns:
A success or failure message indicating if the resource was successfully added.
Throws:
java.lang.Exception

---

getResourceView

java.lang.String **getResourceView**(java.lang.String repositoryName,
java.lang.String itemName,
java.lang.String resourceName)
throws java.lang.Exception

Gets the summary of a resource belonging to an item in the specified repository.

Parameters:

repositoryName - The name of the repository where the resource resides.

itemName - The name of the item the resource is attached to.

resourceName - The name of the resource.

Returns:

-If successful, the XML serialzed resource object is returned.
-If unsuccessful, an error message is returned

Throws:

java.lang.Exception

---

getResourcesView

java.lang.String **getResourcesView**(java.lang.String repositoryName,
java.lang.String itemName,
java.lang.String directives)
throws java.lang.Exception

Gets all the summaries of all resources for the an item in the specified repository.

Parameters:

repositoryName - The name of the repository where the resource resides

itemName - The name of the item the resource is attached to

directives - A list of name/value pairs delimited by the SOAP message delimiter. The available directives are:
- Name: 'enableFilter',
Value: 'All'

Returns:

-If successful, the XML serialzed resource object is returned.
-If unsuccessful, an error message is returned

Throws:

java.lang.Exception

---

updateResource

java.lang.String **updateResource**(java.lang.String repositoryName,
java.lang.String itemName,
java.lang.String resourceName,
java.lang.String crcEnabled,

javax.activation.DataHandler fileAttachment)
throws java.lang.Exception

Updates an existing resource to an item inside the specified repository.

Parameters:

repositoryName - The name of the repository where the item resides.

itemName - The name of the item to update the resource to.

resourceName - The name of the resource to update.

resourceFileCRCCheckingEnabled - A 'true' or 'false' value indicating if the resource file is CRC checking enabled. The default value is 'false'.

fileAttachment - An MTOM attachment containing the resource file.

Returns:

A success or failure message indicating if the resource was successfully updated.

Throws:

java.lang.Exception

---

getResource

javax.activation.DataHandler
getResource(java.lang.String repositoryName,
java.lang.String itemName,

java.lang.String resourceName,

java.lang.String crcEnabled)
throws java.lang.Exception

Gets (downloads) a resource that belongs to an item in the specified repository.

Parameters:

repositoryName - The name of the repository where the item resides.

itemName - The name of the item to get the resource from.

resourceName - The name of the resource.

resourceFileCRCCheckingEnabled - A 'true' or 'false' value indicating if the resource file is CRC checking enabled. The default value is 'false'.

Returns:

The response from FME Server to the getResource command.
-A successful result comes with the resource file as an MTOM attachment.
-An error message is returned if the resource cannot be found or if a fatal exception is thrown when attempting to complete this request.

Throws:

java.lang.Exception

---

removeResource

java.lang.String **removeResource**(java.lang.String repositoryName,
java.lang.String itemName,
java.lang.String resourceName)
throws java.lang.Exception

Removes a resource from an item in the specified repository.

Parameters:

repositoryName - The name of the repository where the item resides.

itemName - The name of the item to which the resource belongs.

resourceName - The name of the resource.

Returns:

A success or failure message based on the result of the resource being removed.

Throws:

java.lang.Exception

---

resourceExists

java.lang.String **resourceExists**(java.lang.String repositoryName,
java.lang.String itemName,
java.lang.String resourceName)
throws java.lang.Exception

Determines if the specified resource exists.

Parameters:

repositoryName - The name of the repository where the item resides.

itemName - The name of the item to which the resource belongs.

resourceName - The name of the resource.

Returns:

'true' if resource exists; otherwise 'false' if resource does not exist

Throws:

java.lang.Exception

---

getGlobalAllItems

java.lang.String **getGlobalAllItems**(java.lang.String enableFilter)
throws java.lang.Exception

Gets all item names from all repositories.

Parameters:

enableFilter - The filter for retrieving global items. The only accepted value is 'All'.

Returns:

-An XML string containing all the item names for each repository

Throws:

java.lang.Exception

---

sessionInit

java.lang.String **sessionInit**(java.lang.String connectionInfo,
java.lang.String credentialType)
throws java.lang.Exception

Attempts to connect to FME Server using the given credentials and credential type.

Parameters:

`connectionInfo` - A string of user ID and password separated by a colon (:).

`credentialType` - A string indicating the credential type. The available types are: - USERID_PASSWORD
- TOKEN

Returns:

- A success message is returned if the connection was successful; otherwise a failure message is returned.

Throws:

`java.lang.Exception`

基本的服务组成要素

- General HTTP Request Rules（常规 HTTP 请求规则）
  - HTTP GET
  - HTTP POST
  - Request Parameters Common to All Services
    - Workspace Published Parameters
    - Transformation Manager Directives
    - FME Engine Directives
- General HTTP Response Rules（常规 HTTP 响应规则）
  - XML Response Schema
- HTTPS
- Authentication

常规 HTTP 请求规则

每个FME Server Web服务都有一个唯一的URL模式以便客户端进行调用操作。当服务安装并启用时，该URL模式定义在FME Server储存库中。所有的服务都支持 HTTP 的 GET 和 POST 方法。 数据上传服务还支持 PUT 方法。

HTTP GET

一个有效的 HTTP GET 请求包含一个 URL 前缀和其它参数来调用 Web 服务。 一个 URL 前缀是指一个包括协议、主机名、可选的端口号、路径、一个问题标记（？）以及选择一个或多个指定的服务参数并以符号（&）结束的字符串。 每个服务实例都有一个唯一的前缀。 路径由一个储存库和一个工作区名称定义。 客户可以以名称/值对 “name=value&”的形式追加额外的请求参数。

下表显示了一个 Web 服务请求的 URL 的组成。

URL Component	Description
http://host[:port]/<service>/<repository>/<workspace>?{name[=value]&}	服务操作的 URL 前缀 []表示 0 or 1 occurrence of an optional part {}表示 0 or more occurrences
name=value&	一个或多个请求参数的名称/值对。 请求参数可以是以下四种类型之一： 服务的指定选项 工作区发布参数 转换管理指令 FME Engine 指令

HTTP POST

客户端可以在 POST 文档正文中编码请求参数和使用 HTTP POST 方法调用服务。如果一个请求包含大量的参数或者参数值比较长，则建议使用 POST 方法。

所有服务共同的请求参数

- Workspace Published Parameters

- Transformation Manager Directives
- FME Engine Directives

常规 HTTP 响应规则

当服务收到有效的请求，它执行指定的操作，并发送回一个响应。 每个 HTTP 响应是伴随着适当的 MIME 类型。 下表描述了服务响应所有可能的 MIME 类型。

Service	Response MIME types
Data Streaming	Dynamically determined by the FME workspace
Data Download	text/xml application/json text/html
Data Upload	text/xml application/json
Job Submitter	text/xml application/json text/html
KML Network Link	application/vnd.google-earth.kmz

HTTPS

除了 HTTP 协议，所有网络服务都支持 HTTPS。使用 HTTPS 协议的服务不会影响上述请求和响应的规则，但它可能需要采取额外的行动通过一个安全连接在客户端和服务间进行通信。

Authentication

所有 Web 服务支持 HTTP 基本身份验证。 当启用 FME Server security 时，服务尝试从请求的 HTTP 头中检索用户名和密码。然后凭据传递给 FME Server 核心建立一个连接。如果认证失败，服务将返回 HTTP 错误代码 401（未经授权）给客户端。

以下是“使用服务 API 示例程序”的 Java 程序示例，它设置 HTTP BASIC 指定请求的凭据。

使用服务 API 示例程序

- Send a GET request to Data Download Service by JavaScript（通过 JavaScript 发送 GET 请求给数据下载服务）
- Send a POST request to the Job Submitter Service using Apache HttpClient in Java（通过 Java Apache HttpClient 发送 POST 请求给任务提交服务）

服务响应示例

- XML 响应示例
- SON 响应示例

JavaScript 程序示例

通过 JavaScript 发送 GET 请求给数据下载服务

下面的代码示例通过使用 XMLHttpRequest 对象发送 GET 请求，并读取响应文本。调用函数 eval （）将文本解析为 JSON 对象。它通过验证服务状态是否成功，如果成功，则打印下载结果的网址。否则打印错误消息。此外，也打印服务器的请求，服务器状态和工作状态的信息。

示例代码

```
var httpRequest = new XMLHttpRequest();
var url =
"http://localhost/fmedatadownload/Samples/austinApartments.fmw?
opt_responseformat=json";
httpRequest.open("GET", url, true);
httpRequest.onreadystatechange = function() {
if (httpRequest.readyState == 4 && httpRequest.status == 200)
{
    var reply = eval("(" + httpRequest.responseText +
")");
    var outputDoc = "Service Response:";
    if (reply.serviceResponse.statusInfo.status == "success")
    {
       var downloadURL =
reply.serviceResponse.url;
       outputDoc += "Service succeeded. Download URL:";
       outputDoc += "<a href=\"" + downloadURL + "\">" +
downloadURL +  "</a>";
    }
    else
    {
       outputDoc+= "Service failed. Error message:";
       outputDoc+= "reply.serviceResponse.service.message;
    }
    outputDoc+= "FME transformation result:";
    outputDoc+= "Request: "
```

```
+   reply.serviceResponse.fmeTransformationResult.fmeServerResp
onse request;
    outputDoc+= "Status message: " +
reply.serviceResponse.fmeTransformationResult.fmeServerResponse
statusMessage;
    outputDoc+= "Job status: " +
reply.serviceResponse.fmeTransformationResult.fmeServerResponse
jobStatus;


    var myWindow = window.open('','','width=800,height=400');
    myWindow.document.write(outputDoc);
    myWindow.focus();
}
```

Java 程序示例

用 Java Apache HttpClient 发送 POST 请求给任务提交服务

如下示例代码创建一个 POST 请求，并通过 HttpClient 发送该请求到 Job Submitter Service。该示例代码还通过 HTTP BASIC 身份验证发送凭据。该请求指定响应格式为 XML 和 TM 指令' tm_priority '为 50。该程序在最后打印出 XML 响应

示例代码

```
public static void main(String[] args)
  {
     HttpClient client = new HttpClient();
      client.getState().setCredentials(new AuthScope("localhost",
8080, "FME Server
Authentication"), newUsernamePasswordCredentials("user",
"user"));
```

```
    String fmeUrl =
"http://localhost:8080/fmejobsubmitter/Samples/austinApartments
.fmw";
    PostMethod method = new PostMethod(fmeUrl);

    method.setDoAuthentication(new);
    // Specify the response format to XML
    method.addParameter("opt_responseformat", "xml");
    // Specify the job priority is 50
    method.addParameter("tm_priority", "50");
    try
    {
       int ret = client.executeMethod(method);
       if (ret == 200) // HTTP 200 OK
       {
          System.out.println(method.getResponseBodyAsString());
       }
    }
    catch (HttpException e)
    {
       System.out.println(e.getStackTrace());
    }
    catch (IOException e)
    {
       System.out.println(e.getStackTrace());
    }
 }
```

XML 响应示例

以下展示了一个完整的以 XML 格式响应数据下载服务

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<serviceResponse>
  <statusInfo>
    <status>success</status>
    <mode>sync</mode>
```

```
  </statusInfo>
  <fmeTransformationResult>
    <fmeEngineResponse>
      <outputLocation>
        C:\Program Files\Apache Software Foundation\Tomcat
5.5\webapps\ROOT\fmedatadownloadresults\FME_3320_1246037797730_
3320871476699.zip
      </outputLocation>
      <resultPrefix>http:</resultPrefix>
      <statusNumber>0</statusNumber>
      <logFileLocation>C:/Program
Files/FMEServer/Logs/datadownload/FME_3320_1246037788246_332087
1476708.log</logFileLocation>
      <numFeaturesOutput>49</numFeaturesOutput>
      <statusMessage>Translation Successful</statusMessage>
      <resultRootDir>//KENOBI:8080/fmedatadownloadresults</resu
ltRootDir>
    </fmeEngineResponse>
    <fmeServerResponse>
      <serviceSuccess>true</serviceSuccess>
      <serviceMsg>success</serviceMsg>
      <id>54</id>
      <resultSuccess>true</resultSuccess>
      <timeFinished>Fri-26-Jun-2009 10:36:37 AM</timeFinished>
      <requesterHost>Kenobi</requesterHost>
      <jobStatus>SUCCESS</jobStatus>
      <result>
        0:Translation Successful|OutputLocation=C:\Program
Files\Apache Software Foundation\Tomcat
5.5\webapps\ROOT\fmedatadownloadresults\FME_3320_1246037797730_
3320871476699.zip|ResultPrefix=http:|ResultRootDir=//KENOBI:808
0/fmedatadownloadresults|NumFeaturesOutput=49|LogFileLocation=C
:/Program
Files/FMEServer/Logs/datadownload/FME_3320_1246037788246_332087
1476708.log
      </result>
      <description>null</description>
      <timeRequested>Fri-26-Jun-2009 10:36:28 AM</timeRequested>
```

```
      <requesterResultPort>3172</requesterResultPort>
      <request>
        "Samples/austinApartments/austinApartments.fmw"
-XML_PARAMETER &lt;?xml version="1.0"
encoding="ISO-8859-1"?&gt;&lt;TRANSFORMATION_REQUEST&gt;&lt;PUB
LISHED_PARAMETER
name="--OPT_responseFormat"&gt;XML&lt;/PUBLISHED_PARAMETER&gt;&
lt;FME_DIRECTIVE
name="-OGCKML_1_DATASET"&gt;!FME_AUTO_DIR_NAME!\austinApartment
s.kml&lt;/FME_DIRECTIVE&gt;&lt;/TRANSFORMATION_REQUEST&gt;
      </request>
      <priority>100</priority>
      <timeStarted>Fri-26-Jun-2009 10:36:28 AM</timeStarted>
      <requestKeyword>FILE_DOWNLOAD_SERVICE</requestKeyword>
    </fmeServerResponse>
</fmeTransformationResult>
<url>http://KENOBI:8080/fmedatadownloadresults/FME_3320_1246037
797730_3320871476699.zip</url>
  <jobID>54</jobID>
</serviceResponse>
```


JSON 响应示例

以下展示了一个完整的以 JSON 格式响应数据下载服务


```
{"serviceResponse": {
   "fmeTransformationResult": {
      "fmeEngineResponse": {
         "logFileLocation": "C:/Program
Files/FMEServer/Logs/datadownload/FME_3320_1246037788246_332087
1476708.log",
         "numFeaturesOutput": 49,
         "outputLocation": "C:\\Program Files\\Apache Software
Foundation\\Tomcat
5.5\\webapps\\ROOT\\fmedatadownloadresults\\FME_3320_1246037797
730_3320871476699.zip",
         "resultPrefix": "http:",
```

```
        "resultRootDir":
"//KENOBI:8080/fmedatadownloadresults",
        "statusMessage": "Translation Successful",
        "statusNumber": 0
    },
    "fmeServerResponse": {
        "description": null,
        "id": 54,
        "jobStatus": "SUCCESS",
        "priority": 100,
        "request":
"\"Samples/austinApartments/austinApartments.fmw\"
-XML_PARAMETER &lt;?xml version=\"1.0\"
encoding=\"ISO-8859-1\"?&gt;&lt;TRANSFORMATION_REQUEST&gt;&lt;P
UBLISHED_PARAMETER
name=\"--OPT_responseFormat\"&gt;XML&lt;/PUBLISHED_PARAMETER&gt
;&lt;FME_DIRECTIVE
name=\"-OGCKML_1_DATASET\"&gt;!FME_AUTO_DIR_NAME!\\austinApartm
ents.kml&lt;/FME_DIRECTIVE&gt;&lt;/TRANSFORMATION_REQUEST&gt;",
        "requestKeyword": "FILE_DOWNLOAD_SERVICE",
        "requesterHost": "Kenobi",
        "requesterResultPort": 3172,
        "result": "0:Translation
Successful|OutputLocation=C:\\Program Files\\Apache Software
Foundation\\Tomcat
5.5\\webapps\\ROOT\\fmedatadownloadresults\\FME_3320_1246037797
730_3320871476699.zip|ResultPrefix=http:|ResultRootDir=//KENOBI
:8080/fmedatadownloadresults|NumFeaturesOutput=49|LogFileLocati
on=C:/Program
Files/FMEServer/Logs/datadownload/FME_3320_1246037788246_332087
1476708.log",
        "resultSuccess": true,
        "serviceMsg": "success",
        "serviceSuccess": true,
        "timeFinished": "Fri-26-Jun-2009 10:36:37 AM",
        "timeRequested": "Fri-26-Jun-2009 10:36:28 AM",
        "timeStarted": "Fri-26-Jun-2009 10:36:28 AM"
    }
```

```
  },
  "jobID": 54,
  "statusInfo": {
    "mode": "sync",
    "status": "success"
  },
  "url":
"http://KENOBI:8080/fmedatadownloadresults/FME_3320_12460377977
30_3320871476699.zip"
}}
```