# Introduction

This document outlines the software requirements of the Pill build system.

## Goals

## Must Have

* Scripts written in Java
* Distributed servers
* Built modules indicate where they got their dependencies from

## Nice to Have

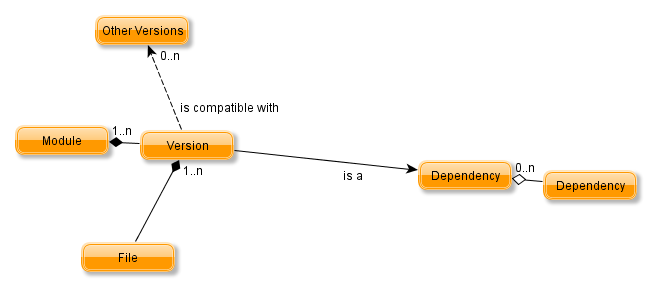
* Support any version scheme
* Avoid executing untrusted code
* Ability to serve modules off code repositories, which do not implement our custom RESTful protocol

## Bootstrapping the Build

* Both Ant and Maven build themselves if a bootstrap binary exists.
* Ant uses batch files to build the bootstrap binary. Maven uses Ant to build the bootstrap binary.
* We’ll use Ant to build the bootstrap.

## Components

The system consists of the following components:



A **Module** consists of one or more **Releases**. Each release consists of one or more **Files** and depends on zero or more **Dependencies**.

## Server Communication Protocol

The data server uses a RESTful[[1]](#footnote-1) communication protocol that is built around the transfer of representations of resources. A resource can be essentially any coherent and meaningful concept that may be addressed. A representation of a resource is typically a document that captures the current or intended state of a resource. For example, Facebook pages and images embedded therein are may be represented as resources that may be uploaded or downloaded by web clients (browsers).

All resources may respond with:

* **401 Unauthorized** if the client is not authorized to issue the request
* **415 Unsupported Media Type** if the content-type of the request body is not supported

The following JSON types are defined:

* **integer** is a number from -231 to 231-1
* **string** is a collection of characters
* **date** is date without a time formatted according to [ISO-8601](http://en.wikipedia.org/wiki/ISO_8601)
* **time** is time without a date formatted according to [ISO-8601](http://en.wikipedia.org/wiki/ISO_8601)
* **dateTime** is a combination of a date and time according to [ISO-8601](http://en.wikipedia.org/wiki/ISO_8601)
* **base64** is a text encoding scheme used to represent binary data
* **url** is the location of a resource
* **urn:repository** is the URL relative to which all resource paths are defined.

### Modules

#### Create a Module

|  |  |
| --- | --- |
| **Client Request** | POST /modules |
| **Request Headers** | **Content-type**: application/x-org.pill.modules+json; version=1.0 |
| **Request Body** | {  “name”: **name**  } |
| **Response Parameters** | See [GET **urn:*module***](#_Get_a_Module’s) |
| **Response Code** | **201 Created** |
| **Response Headers** | **Location**: the canonical **urn:module** of the new resource |
| **Response Body** |  |

|  |  |
| --- | --- |
| **Response Code** | **400 Bad Request** if the request body would leave the resource in an incomplete or inconsistent state. |
| **Response Headers** | **Content-type**: text/plain |
| **Response Body** | A user-friendly explanation of why the request is invalid |
| **Use Cases** | **name** was an empty string |

|  |  |
| --- | --- |
| **Response Code** | **409 Conflict** if another module has the same name. |
| **Response Headers** | **Content-type**: text/plain  **Location**: **urn:module** of the existing module |
| **Response Body** | A user-friendly explanation of why the request is invalid |

#### Get a Module’s State

|  |  |  |  |
| --- | --- | --- | --- |
| **Client Request** | GET **urn:module** | | |
| **Request Headers** |  | | |
| **Request Body** |  | | |
| **Response Code** | **200 Ok** on success | | |
| **Response Headers** | **Content-type**: application/x-org.pill.module+json; version=1.0 | | |
| **Response Body** | {  “repository”: **repository**,  “name”: **name**,  “releases”:  [  **releases**  ]  } | | |
| **Response Parameters** | **Name** | **Type** | **Description** |
| repository | urn:repository | The repository this module came from. |
| name | string | The module’s name. |
| releases | urn:release | The module’s releases. |

|  |  |
| --- | --- |
| **Response Code** | **404 Not Found** if the module does not exist |
| **Response Headers** |  |
| **Response Body** |  |

#### Modify a Module

|  |  |
| --- | --- |
| **Client Request** | PUT **urn:module** |
| **Request Headers** | **Content-type**: application/x-org.pill.module+json; version=1.0 |
| **Request Body** | Same as request body of [POST /modules](#_Create_a_Clinic) |
| **Response Code** | **200 Ok** if the module was modified |
| **Response Headers** |  |
| **Response Body** |  |

|  |  |
| --- | --- |
| **Response Code** | **400 Bad Request** if the request body would leave the resource in an incomplete or inconsistent state |
| **Response Headers** | **Content-type**: text/plain |
| **Response Body** | A user-friendly explanation of why the request is invalid |

|  |  |
| --- | --- |
| **Response Code** | **404 Not Found** if the specified module does not exist |
| **Response Headers** |  |
| **Response Body** |  |

#### List Modules

|  |  |  |  |
| --- | --- | --- | --- |
| **Client Request** | GET /modules | | |
| **Request Headers** | **Content-type**: application/x-org.pill.modules+json; version=1.0 | | |
| **Request Body** |  | | |
| **Response Code** | **200 Ok** on success | | |
| **Response Headers** | **Content-type**: application/json; version=1.0 | | |
| **Response Body** | A map of module names to their address:  {  **name**: **module**  } | | |
| **Response Parameters** | **Name** | **Type** | **Description** |
| name | string | The module’s name. |
| module | urn:module | The module’s location. |

#### Remove a Module

|  |  |
| --- | --- |
| **Client Request** | DELETE **urn:module** |
| **Request Headers** | **Content-type**: application/x-org.pill.module+json; version=1.0 |
| **Request Body** |  |
| **Response Code** | **200 Ok** |
| **Response Headers** |  |
| **Response Body** |  |

#### Releases

##### Create a Release

|  |  |
| --- | --- |
| **Client Request** | POST **urn:release** |
| **Request Headers** | **Content-type**: application/x-org.pill.release+json; version=1.0 |
| **Request Body** | {  “module”: **module**,  “version”: **version**,  “files”:  [  **files**  ],  “dependencies”:  [  {  “module”: **dependencyModule**,  “version”: **dependencyVersion**,  “type”: **type**,  “uri”: **uri**  }  ]  } |
| **Response Parameters** | Same as response body of [GET **urn:release**](#_Get_a_Doctor's) |
| **Response Code** | **201 Created** |
| **Response Headers** | **Location**: the canonical **urn:release** of the new resource |
| **Response Body** |  |

|  |  |
| --- | --- |
| **Response Code** | **400 Bad Request** if the request body would leave the resource in an incomplete or inconsistent state |
| **Response Headers** | **Content-type**: text/plain |
| **Response Body** | A user-friendly explanation of why the request is invalid |
| **Use Cases** | The version name is an empty string |

|  |  |
| --- | --- |
| **Response Code** | **409 Conflict** if another version or dependency has the same name. |
| **Response Headers** | **Content-type**: text/plain  **Location**: **urn:release** of the existing version |
| **Response Body** | A user-friendly explanation of why the request is invalid |

##### Get a Release’s State

|  |  |  |  |
| --- | --- | --- | --- |
| **Client Request** | GET **urn:release** | | |
| **Request Headers** |  | | |
| **Request Body** |  | | |
| **Response Code** | **200 Ok** on success | | |
| **Response Headers** | **Content-type**: application/x-org.pill.release+json; version=1.0 | | |
| **Response Body** | {  “module”: **module**,  “version”: **version**,  “files”:  [  **file**  ],  “dependencies”:  [  {  “module”: **dependencyModule**,  “version”: **dependencyVersion**,  “type”: **type**,  “uri”: **uri,**  “hash”: **dependencyHash**  }  ],  “hash”: **hash**  } | | |
| **Response Parameters** | **Name** | **Type** | **Description** |
| module | urn:module | The module associated with this release. |
| version | string | The version number associated with this release. |
| file | urn:file | A file belonging to the release. |
| dependencyModule | string | The module name of the dependency |
| dependencyVersion | string | The version of the dependency module. |
| type | string | The dependency type. One of [“runtime”, “build”]. “runtime” denotes runtime dependencies required to use the library. “build” denotes any other kind of dependency (for example, dependencies used to build or test the module). |
| uri | urn:release | The URI of the dependency. |
| dependencyHash | base64 | The dependency’s hash value. See “hash” below. |
| hash | base64 | A sha-1 hash of [module.name, version, file.hash, dependencyModule, dependencyVersion, type, uri]. All Strings are encoded using UTF-8. Releases with different hash values are guaranteed to differ (the converse is not necessarily true). |

|  |  |
| --- | --- |
| **Response Code** | **404 Not Found** if the version does not exist |
| **Response Headers** |  |
| **Response Body** |  |

##### Remove a Release

|  |  |
| --- | --- |
| **Client Request** | DELETE **urn:release** |
| **Request Headers** | **Content-type**: application/x-org.pill.release+json; version=1.0 |
| **Request Body** |  |
| **Response Code** | **200 Ok** |
| **Response Headers** |  |
| **Response Body** |  |

##### Files

###### Create a File

|  |  |
| --- | --- |
| **Client Request** | POST **urn:release** |
| **Request Headers** | **Content-type**: application/x-org.pill.release.file+json; version=1.0 |
| **Request Body** | Same as response body of [PUT **urn:file**](#_Modify_a_File) |
| **Response Code** | **201 Created** |
| **Response Headers** | **Location**: the canonical **urn:file** of the new resource |
| **Response Body** |  |

|  |  |
| --- | --- |
| **Response Code** | **400 Bad Request** if the request body would leave the resource in an incomplete or inconsistent state |
| **Response Headers** | **Content-type**: text/plain |
| **Response Body** | A user-friendly explanation of why the request is invalid |
| **Use Cases** | The file name is an empty string. |

|  |  |
| --- | --- |
| **Response Code** | **409 Conflict** if another file has the same name. |
| **Response Headers** | **Content-type**: text/plain  **Location**: **urn:file** of the existing file |
| **Response Body** | A user-friendly explanation of why the request is invalid |

###### Get a File’s Metadata

|  |  |  |  |
| --- | --- | --- | --- |
| **Client Request** | GET **urn:file** | | |
| **Request Headers** |  | | |
| **Request Body** |  | | |
| **Response Code** | **200 Ok** on success | | |
| **Response Headers** | **Content-type**: application/x-org.pill.release.file+json; version=1.0 | | |
| **Response Body** | {  “release”: **release**,  “name”: **name**,  “type”: **type**,  “hash”: **hash**,  “data”: **data**  } | | |
| **Response Parameters** | **Name** | **Type** | **Description** |
| release | urn:release | The release this file is associated with. |
| name | string | The file name. |
| type | string | The file type. |
| hash | base64 | A sha-1 hash of [data]. Files with different hash values are guaranteed to differ (the converse is not necessarily true). |
| data | urn:fileData | The file contents. |

|  |  |
| --- | --- |
| **Response Code** | **404 Not Found** if the file does not exist |
| **Response Headers** |  |
| **Response Body** |  |

###### Modify a File’s Metadata

|  |  |  |  |
| --- | --- | --- | --- |
| **Client Request** | PUT **urn:file** | | |
| **Request Headers** | **Content-type**: application/x-org.pill.release.file+json; version=1.0 | | |
| **Request Body** | {  “name”: **name**,  “type”: **type**,  } | | |
| **Request Parameters** | **Name** | **Type** | **Description** |
| name | string | The file name. |
| type | string | The file type. |
| **Response Code** | **200 Ok** if the file was modified | | |
| **Response Headers** |  | | |
| **Response Body** |  | | |

|  |  |
| --- | --- |
| **Response Code** | **400 Bad Request** if the request body would leave the resource in an incomplete or inconsistent state |
| **Response Headers** | **Content-type**: text/plain |
| **Response Body** | A user-friendly explanation of why the request is invalid |

|  |  |
| --- | --- |
| **Response Code** | **404 Not Found** if the specified file does not exist |
| **Response Headers** |  |
| **Response Body** |  |

###### Get a File’s Contents

|  |  |
| --- | --- |
| **Client Request** | GET **urn:fileData** |
| **Request Headers** |  |
| **Request Body** |  |
| **Response Code** | **200 Ok** on success |
| **Response Headers** | **Content-type**: application/octet-stream; version=1.0 |
| **Response Body** | The file’s contents |

|  |  |
| --- | --- |
| **Response Code** | **404 Not Found** if the file does not exist |
| **Response Headers** |  |
| **Response Body** |  |

###### Set a File’s Contents

|  |  |
| --- | --- |
| **Client Request** | PUT **urn:fileData** |
| **Request Headers** | **Content-type**: application/octet-stream; version=1.0 |
| **Request Body** | The file’s new contents. |
| **Response Code** | **200 Ok** if the file was modified |
| **Response Headers** |  |
| **Response Body** |  |

|  |  |
| --- | --- |
| **Response Code** | **400 Bad Request** if the request body would leave the resource in an incomplete or inconsistent state |
| **Response Headers** | **Content-type**: text/plain |
| **Response Body** | A user-friendly explanation of why the request is invalid |

|  |  |
| --- | --- |
| **Response Code** | **404 Not Found** if the specified file does not exist |
| **Response Headers** |  |
| **Response Body** |  |

###### Remove a File

|  |  |
| --- | --- |
| **Client Request** | DELETE **urn:file** |
| **Request Headers** | **Content-type**: application/x-org.pill.release.file+json; version=1.0 |
| **Request Body** |  |
| **Response Code** | **200 Ok** |
| **Response Headers** |  |
| **Response Body** |  |

1. http://en.wikipedia.org/wiki/Representational\_State\_Transfer [↑](#footnote-ref-1)