**Video 5**

* **Register New User URI**

Complete URL address: <http://www.yourWebSiteAddress.com/ai/users/?start=1&limit=50>

* http : Protocol
* www: SubDomain
* yourWebSiteAddress.com: Website address
* ai/users/ : URI
* start=1&limit=50 : Query String

**HTTP methods**

* **Example of traditional Post method**

POST /RegisterUserServiet

{

"firstName": "James",

"lastName":"Bond"

}

* **Example of Rest Post method**

POST /users

{

"firstName": "James",

"lastName":"Bond"

}

**Other HTTP Methods**

* **Traditional**

GET /GetUserDetailsServlet?userld=1

GET /DeleteUserDetailsServlet?userld=1

POST /UpdateUserDetailsServlet

"firstName": "James",

"lastName":"Bond"

"userld" : "1"

}

**Rest**

GET /users/1

DELETE /users/1

PUT /users/1

{

“firstName" ; "James",

“lastName” : “Bond”

}

**Longer URIs**

* **Traditional REST**

/GetMessageComments?userid=1&messageld=5

/GetMessageComment?userld=1&messageld=5&commentid=1

* **Rest**

GET /use,s/1/messages/5/comments

GET /users/1/messages/S/comments/1

**Naming Tips**

* Nouns as opposed to verbs or actions. Ex. /users vs /getUsers/
* Plurals. Ex /users/1 vs /user/1
* Predictable.
  + Ex. /users/1/messages rather than /users/1/publishedMessages/
  + /users/1/messages/2/likes rather than /users/1/publishedMessages/2/allLikes

--------------------------------------------------------------------------------------------------------------------------------------

**Video 6**

**HTTP Methods**

* **Non RESTful**

POST - Create user. /RegisterUserServiet

GET - Read user details. /GetUserDetailsServiet?userld=1

POST - Update user details. /UpdateUserDetailsServiet

GET - Delete user details. /DeleteUserDetailsServiet?userid=1

* **RESTful**

POST - Create user. /users

GET - Read user details. /users/1

PUT - Update user details. /users/1

DELETE - Delete user details. /users/1

**Video 7**

**Headers: Accept and Content Type**

**Content Type:** To be sent to the service site

* Application JSON: The type of content format we send to the service site when we Post
* Accept: This is the type of content we expect back from the service site, and it is also in application JSON format