|  |  |
| --- | --- |
| Requirement | Implementation |
| 1. A blog author is able to register to the platform | Method: POST, URI: /blog/users  Body: { “email”, [eddy@yahoo.com](mailto:eddy@yahoo.com), “username”: “eddy”, “password”: “pass123” } Body  Returns sessionId  Handler: com.epc.blog.CreateUserHandler  Method: POST, URI: /blog/users/{userId}  Body: { “password”: “passwordHash” }  Returns sessionId  sessionId is a security token associated with a user. A table maps this sessionId against a user.  Handler: com.epc.blog.UserSessionHandler |
| 1. A blog author may create a space to publish their blogs | Method: POST, URI: /blog/spaces  Body: { “sessionId”: “sessionId”, “uri”: “path1/path\_to\_this\_space” }  Returns spaceId  The server will get the user associated with the sessionId. Then a new space will be created and owned by the said user.  Handler: com.epc.blog.CreateBlogSpaceHandler |
| 1. A blog author can grant access to other authors to publish to their space | Method: PATCH, URI: /blog/spaces/{spaceId}  Body: { “userIds”: [ “mary”, “john” ], “sessionId”: “sessionId” }  Returns None  The server will make sure that the user associated with the sessionId is in fact the owner of the space. Return http code 403 if user does not own the space.  Handler: com.epc.blog.GrantUserAccessHandler |
| 1. A blog author must approve any blog entries published in their space | Method: PUT, URI: /blog/blogEntries/{blogEntryId}  Body: { “approval”: true, “sessionId”: “sessionId” }  Returns None  Handler: com.epc.blog.ApproveBlogEntryHandler |
| 1. A registered author can create a new blog entry in their own, or another author’s space | Method: POST, URI: /blog/blogEntries  Body: { “spaceId”: “spaceId”, “sessionId”: “sessionId”, “uri”: “path2/path\_to\_this\_entry }  Returns the blogEntryId  The server will check whether the user associated with the sessionId has permissions on the space  Handler: com.epc.blog.CreateBlogEntryHandler |
| 1. A registered author can fetch a list of their blog entries, select one and view. | Method: GET, URI: /blog/blogEntries?sessionId={sessionId}  Returns : A list of blogIds, and URIs [ {“blogEntryId”: “123”: “uri”: “resource/path/1”}, {“blogEntryId”: “234”, “uri”, “resource/path/2”, “approved”: “false”… ]  After the user chooses the blog entry he is interested in, the URI associated with his selection can be used to retrieve the blog entry.  Handler: com.epc.blog.FindUserBlogEntriesHandler |
| 1. Any user can view and read blog entries in any space. | Method: GET, URI: /blog  Returns a list which contain the following:  [ {“spaceId”: “123”, “uri”: “uri1”, “blogEntries”: [ {“blogEntryId”: “1234”, “uri”: “uri2”}…]},  {“spaceId”: “124”, “uri”: “uri1”, “blogEntries”: [ {“blogEntryId”: “1235”, “uri”: “uri2”}…]},…]  Handler: com.epc.blog.FindBlogsHandler |