

# 1 Access Channel Requirements

When considering the access channels of a target system, there are two main possibilities to consider; the human users of the system and whether any other systems need to be able to access the system. For both of these possibilities one then needs to consider through which ways they will access the system, what functionality the system requires these access channels to have and how these access channels will provide said functionality.

The system required by our client (will from herewith be referred to as the system or the target system) is a system through which researchers can keep track of their publications. The users of the system, the researchers, need a channel through which they can use the system. Our client does not require any other systems to be able to access our system or use the functionality of our system for a higher processing scheme, at this stage.

For our human users of the system, there are two platforms, our client would like them to be able to access the system by:

- A Web Interface
- A Mobile Application (On the Android platform)

Both of these will have to access our system through the Internet, for the Android Application this will require requesting permissions. The abovementioned platforms will need an interface through which to access our system or what is known as an Application Programming Interface (API), this interface will have to enable the passing of data from the system to the channel our user accesses the system through. Considering the general functionality our system provides, and that communication would take place through the Internet this API would have to be RESTful.

REST stands for Representational State Transfer, and it is a software architectural style that deals with resources and what resources are accessed through using the HTTP user-oriented network protocol. This RESTful API would enable the transfer of data, between our system and the platforms our client has chosen for users to access the system through, over a network, in our case the Internet.

Before we consider this API further and what data it shall pass and how our access channels will use the API, let's revisit the functionality the access platforms will need to have to provide the users with the target systems maximum functionality.

- User log in
- Search functionality
- Ability to open excel spreadsheets
- Viewing of plain text
- Viewing of lists, tables etc.

- Viewing of links
- Ability to view information
- Ability to add information
- Ability to change information
- Ability to remove information

For the viewing of text, lists, links and the like, this will have to do with the Graphical User Interface (GUI) that the access channel provides. In the case of the Web Interface this can be designed using HTML and its associated counterparts such as CSS, bootstrap and more. In the case of the Android Application, a number of classes and functions will be used. For example in the case of a list, one would use a ListView item on their user interface and this can be created using the android.widget.ListView class, and operations on said list will be done using the classes associated functions.

The Search functionality would also need to be provided by the Graphical User Interface, but actual searching of the database used in our system will be done by the system itself, so our API would need to be able to request the system to do the search, when a user provides the required information for the search by interacting with the GUI.

For the ability to open excel spreadsheets, in the Web Interface, this would imply the ability to download a file while on an Android Application this would imply the same thing as well as the improved ability to open the user devices spreadsheet viewer.

Now for the user log in functionality, users would provide user details through the GUI and the API would then need to pass user details for verification to the system and be able to receive a response about whether or not log in was successful or not. Another option that would be safer is the use of API authentication, this is a token based authentication in which the user would log in and if successful the system will respond with a unique token that can be used in future requests.

As we already know the information that will be displayed on these platforms GUI's will be provided through the RESTful API from our system. For this information, the API would need to get information from the system and be able to send changes including the adding and removing of information back to the system. Basically all the interaction between the system and these platforms will be done by the API with the following HTTP methods:

- GET – For reading information from the system
- PUT – For adding information to the system
- DELETE – For removing information from the system
- POST – For making changes/edits to information on the system

- OPTIONS – For getting operations that can be performed by the system

While these platforms vary, they will both access the system through this API and use the Internet to communicate with the system. However the ways they call the API, process and deal with the responses will vary slightly on the two platforms. The API would provide a standard way for accessing the system despite what channel may be in use. These access channels thus simply need to provide users with the functionality that the target system requires on a GUI and use the API for providing users with the results and information from the system.

### *References*

- <http://www.andrewhavens.com/posts/20/beginners-guide-to-creating-a-rest-api/>
- [http://www.tutorialspoint.com/restful/restful\\_introduction.htm](http://www.tutorialspoint.com/restful/restful_introduction.htm)