

Wrist Steam – System Manual

Authored by Austin Hull

Table of Contents:

1. Manual Use	p. 2
2. System Module Overview	p. 3
3. Configuration Module	p. 4
4. Steam API Module	p. 5
5. User Interface/Information Display Module	p. 6

1. Manual Use:

This manual is intended for those interested in gaining a better understanding of the Wrist Steam application architecture, and the various ways in which its systems interact and operate. Note that the Wrist Steam system involves multiple programming frameworks, including the Pebble SDK (C), PebbleKitJS (Javascript), and node.js (Javascript). This manual will assume that the user has an adequate understanding of the features and usage of these frameworks. For end-user-specific information and guidance regarding the configuration and operation of this application, please refer to the **Wrist Steam – User Manual** documentation.

2. System Module Overview:

The Wrist Steam system is comprised of multiple modules:

Configuration Module: This module allows the user to configure the Wrist Steam application with their Steam user profile. This module interacts with Steam's OpenID authentication service in order to securely retrieve profile metadata, and does this through a node.js server application.

Steam API Module: This module utilizes the user's mobile device in order to retrieve the user's relevant profile data through web requests made to the Steam API. The retrieved information is sent to the **User Interface/Information Display Module** for further processing.

User Interface/Information Display Module: This module takes information received from the **Steam API Module** and processes it to be displayed on the Pebble smartwatch display. After processing, the data will be applied to the appropriate menu fields, which can be accessed, viewed, and scrolled through by the user.

3. Configuration Module:

Description: The **Configuration Module** is responsible for allowing the system user to link their Steam user profile (and potentially other user-related information and preferences) to the Wrist Steam application.

The typical operation of this module follows the following basic process:

1. User selects the “settings” button for the Wrist Steam smartwatch application within the Pebble phone application.
2. This will activate the Javascript side of the Wrist Steam system, opening a webpage directed to a node.js server application. This server page allows the user three actions – to log into Steam in order to retrieve their profile ID number, to save and exit the server page in order to finalize the completed configuration, or to exit the server page without saving any retrieved configuration information. Saved configuration information will be saved through the Javascript side of the system.
3. When the user clicks on the “Log in Through Steam” button, the node.js server will redirect the user to Steam’s official OpenID authentication page.
4. User will log into Steam through the official OpenID authentication page.
5. Upon successful login, user will be returned to the node.js server page, along with the Steam profile’s ID number.
6. Clicking the save button on the node.js server page will forward the profile ID number to the Javascript side of the Wrist Steam application to be saved for system use.
7. When the application is restarted, the **Steam API Module** will process the user’s profile ID number through various web requests to the Steam API in order to retrieve relevant data.

4. Steam API Module:

Description: The **Steam API Module** is contained within the Javascript side of the Wrist Steam application, and is responsible for making web requests to the Steam API services, which in-turn return relevant user data and other associated classes of information. These web requests are made via XMLHttpRequest calls, and utilize the user's Steam profile's ID number returned via the **Configuration Module**.

Typical operation of this module follows the following basic process:

1. Information is saved to the Javascript side of the Wrist Steam system via the **Configuration Module**. This information primarily refers to the user's Steam profile ID number.
2. Application is restarted by user to complete configuration process.
3. Module makes an XMLHttpRequest using an appropriate API web address, including a supplied website key (generated by Steam's API services) as well as the now-saved Steam profile ID number.
4. Requested data structures are returned to the application in the JSON format. Each relevant data object will be parsed from the overall data structure, and will then be placed in its own Javascript variable for later use. Note that data retrieved representing the user's friend's profile names may be concatenated into a single variable to be parsed later within the smartwatch side of the application.
5. After all data objects have been allocated to their proper variables, they will be sent to the smartwatch side of the application via a **dictionary** data structure, which carries data from the Javascript side of the application to the smartwatch side of the application in the same order that **KEY_** definitions are declared near the top of the **main.c** file (which belongs to the smartwatch side of the application).

5. User Interface/Information Display Module

Description: The **User Interface/Information Display Module** makes up a majority of the smartwatch side of the Wrist Steam application. This module is responsible for receiving and processing data received from the Javascript side of the application. In addition to this, it is also responsible for interpreting and responding to user input, as well as displaying appropriately-formatted versions of the aforementioned, processed data. This module also maintains initialization of the application, and keeps track of the various communication connections between the smartwatch, the phone, and the Steam API services.

Typical operation of this module follows the following basic process:

1. User executes application via smartwatch menu. Application renders a splash screen, notifying user that the program is loading and retrieving required data from the Steam API services. If the connection to the phone is nonexistent or lost, or if data is not fully retrieved within 30 seconds, the user will be notified, and may attempt to reload the program by clicking the **SELECT** button on the smartwatch.
2. When program is fully loaded and appropriate user data is fully retrieved from Steam API services, the splash screen is replaced by the application's main menu window. At this point, the splash screen is removed from the window hierarchy of the application, and deleted from memory, thus freeing any memory resources reserved for said process.
3. From the main menu window, the user may select between the various displays of the application, including displays of their own profile data, their friend's profile names, and guide information along with "About" information regarding the application itself. These displays are all navigated through using the **BACK**, **SELECT**, **UP**, and **DOWN** buttons of the Pebble smartwatch.
4. When the user is done using the application, they can hit the **BACK** button as many times as required to return to the menu of the smartwatch system itself. Doing this will free all memory allocated to the Wrist Steam application, thus allowing any program resources used to be available for the next application executed.