

File Index – This file serves to list the files that exist in the various parts of the Off The Rails project and to provide an explanation of what each one does. This ‘index’ is broken up into three major directories: Server, Client, and Mobile Cab (Alternative Client). In each directory area, the files need only be left in the main project directory to function as expected.

| Server | |
|-------------------|--|
| Filename | Purpose |
| DerailedRail.cs | This file provides a class definition for a rail that a train has derailed on within the simulation. |
| frmControl.cs | This is the Windows Form that launches when the server application starts. It also launches an instance of the Interface class upon startup, and cleans it up when the form closes. This file is linked with the frmControl.designer.cs file, but that is auto-generated so there is no need to list it here. |
| Interface.cs | This file provides the class definition for the Interface class, which sets up the HTTP request-based API that the server uses to communicate with its clients. |
| Logger.cs | This file provides the class definition for the Logger class. This class is used to provide a logging Windows Forms listBox that can be written to from any thread, not just the main UI thread. |
| Options.cs | This file provides the class definition for the Options class. This class is used to keep track of the settings for the simulator that the user selects on the client-side index.html. |
| Program.cs | This is the starting point of the server application. All it does is simply launch an instance of FrmControl to begin the server. |
| Rail.cs | This file provides a class definition for the Rail class, which is an important component of running the simulation. |
| Simulator.cs | This file provides the class definition of the Simulator class. This class handles the lion’s share of running the simulation, as can probably be guessed by its name. Each instance of this class contains lists of Trains, Rails, and Turnouts & runs updates on them on a timer according the options sent to it via an Options object. |
| Train.cs | This file provides a class definition for the Train class. This class describes what the meaning of a Train is in the simulation. |
| Turnout.cs | This file provides a class definition for the Turnout class. This class describes how a turnout works in the simulation. |
| WebServerRoute.cs | This file provides a class definition for the WebServerRoute class, which is used to set up HTTP request API endpoints. Each endpoint in the API consists of a WebServerRoute instance. |

| Client | |
|----------------|--|
| Filename | Purpose |
| connections.js | This JavaScript file provides a host of methods to use when connecting to the server's API. It is referenced by both index.html and view.html. Almost any method you see being used in the other JavaScript files in this directory that sounds like it deals with a server action (e.g., 'getTrains()') is likely defined within this file. |
| index.html | This is the front page for the client-side portion of this project. After the server portion has been launched, a user is able to start a simulation instance from this web page with a small set of configurable options. When the request to start the server completes successfully, the user is redirected to view.html. |
| rail.js | This file provides the class definition for the client-side Rail class. It contains a Draw function so that a rail can be drawn to the view with a single call. |
| sketch.js | This JavaScript file is the backend for view.html. It uses a lot of the features provided by the p5.js JavaScript graphics library to illustrate the current state of the simulation to the user. |
| train.js | This file provides the class definition for the client-side Train class. It contains a Draw function so that a train can be drawn to the view with a single call. |
| turnout.js | This file provides the class definition for the client-side Turnout class. It contains a Draw function so that a turnout can be drawn to the view with a single call. |
| view.html | This webpage contains the main view where a user can run the simulation. It has a control panel where they can select which train/turnout to control, and it has a canvas that has the current state of the simulation drawn onto it every frame. |

| Mobile Cab (Alternative Client) | |
|---------------------------------|---|
| Filename | Purpose |
| alternateConnection.js | This JavaScript file provides the back-end for alternatePage.html. It contains some startup methods to populate the controls on alternatePage.html and some methods to interact with the server's API. |
| alternatePage.html | This webpage acts as a 'sketch' for what a mobile cab could look like in this system. It is very rough, and more work needs to be done to prevent users from fighting each other for the same resource, but it illustrates the potential of a mobile cab. |