

Request/Response

Request pattern:

```
{  
    command: ~The Command~,           -> Int  
    data: ~Data If Required~           -> JSON Object  
}
```

Response pattern:

```
{  
    command: ~The Command~,           -> Int  
    code: ~Response Code~,            -> Int  
    data: ~Data If Required~          -> JSON Object  
}
```

The list of requests/responses:

// Action: Log in the User

```
{  
    command: LOGIN,  
    data: ~Credentials~                -> Credentials  
}  
  
{  
    command: LOGIN,  
    code: ~ResponseCode~,  
    data: {  
        error: ~Error Message~        -> String  
    }  
}
```

// Action: Add the last created route to the database

```
{  
    command: ADD_ROUTE,  
    data: {  
        total_distance: ~Kilometers~   -> Double  
    }
```

```

        polyline: ~Google Maps Polyline~,           -> String
        stops: [ ~Cities~ ],                         -> City[]
    }
}
{
    command: ADD_ROUTE,
    code: ~Response Code~,
    data: {
        id: ~Route ID~                               -> Int
    }
}

```

// Action: Get a complete information of a route from the database

```

{
    command: GET_ROUTE,
    data: {
        id: ~Route ID~                               -> Int
    }
}
{
    command: GET_ROUTE,
    code: ~Response Code~
    data: {
        route: ~Route~                               -> Route
    }
}

```

// Action: Get a brief list of routes information

```

{
    command: GET_ROUTE_LIST,
    data: { <empty> } // may contain some filters later
}
{
    command: GET_ROUTE_LIST,
    code: ~Response Code~,
    data: {

```

<pre> routes_list: [route: { id: ~Route ID~, info: ~Route Info~ }] } } </pre>	<p>-> JSON array</p> <p>-> JSON object</p> <p>-> Int</p> <p>-> RouteInfo</p>
---	--

Glossary

blue - to be used for the current version

grey - for the future development

Common data types:

Credentials {

name: ~User Name~,	-> String
password: ~User Password~	-> String

}

Route {

id: ~Route ID~,	-> Int
info: ~Short Route Info~,	-> RouteInfo
polyline: ~Google Maps Polyline~,	-> String
stops: [~Cities~]	-> City[]
path:Path; // * _ *	

}

RouteInfo {

origin: ~From City~,	-> City
destination: ~To City~,	-> City
total_distance: ~Kilometers~	-> Double

}

City {

place_id: ~Google Maps Place~	-> String
formatted_address: ~City Name~	-> String
location: ~Location~	-> Location

```

        population: Int;
    }
    Location {
        lat: ~Latitude~           -> Double
        lng: ~Longitude~           -> Double
    }

    Path {                          // *_*
        roads: Road[];
        ???
    }

    Road {                          // *_*
        id: ~Road ID/Place ID~,     -> String
        name: ~Road Name~,          -> String
        // M05 (Kiev-Odessa), M06 (Kiev-Lviv) etc.
        statistics: ~Road Statistics~ -> ???
        // quality, traffic etc.
        ???
    }

```

Response codes:

OK	= 200
BadRequest	= 400
Unauthorized	= 401
Forbidden	= 403
NotFound	= 404

Request codes:

LOGIN	= 0
ADD_ROUTE	= 1
GET_ROUTE	= 2
GET_ROUTE_LIST	= 3

****Add your argued propositions here.****

