

Request/Response

Request pattern:

```
{  
    command: ~The Command~,  
    data: ~Data If Required~  
}
```

-> String
-> JSON Object

Response pattern:

```
{  
    command: ~The Command~,  
    code: ~Response Code~,  
    data: ~Data If Required~  
}
```

-> String
-> Int
-> JSON Object

The list of requests/responses:

// Action: Log in the User

```
{  
    command: "login",  
    data: ~Credentials~  
}  
{  
    command: "login",  
    code: ~ResponseCode~,  
    data: {  
        error: ~Error Message~  
    }  
}
```

-> Credentials
-> String

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

```
{  
    command: "addRoute",
```

```

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

// Action: Get a brief list of routes information
{
    command: "getRouteList",
    data: { <empty> } // may contain some filters later
}
{
    command: "getRouteList",
    code: ~Response Code~,
    data: {
        routes_list: [                             -> JSON array
            route: {                               -> JSON object
                id: ~Route ID~,                     -> Int
                info: ~Route Info~                  -> RouteInfo
            }
        ]
    }
}

```

```

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

```

```

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

```

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~,
        destination: ~To City~,
        total_distance: ~Kilometers~
    }
    City {
        place_id: ~Google Maps Place~
        formatted_address: ~City Name~
        location: ~Location~
        population: Int;
    }
    Location {
        lat: ~Latitude~
        lng: ~Longitude~
    }

    Path {
        roads: Road[];
        ???
    }
    Road {
        id: ~Road ID/Place ID~,
        name: ~Road Name~,
        // M05 (Kiev-Odessa), M06 (Kiev-Lviv) etc.
        statistics: ~Road Statistics~
        // quality, traffic etc.
        ???
    }

```

Response codes:

// TODO

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

