

Request/Response

Request pattern:

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

Response pattern:

```
{  
    command: ~The Command~,           -> String  
    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: "addRoute",  
    data: {  
        total_distance: ~Kilometers~   -> Double  
        polyline: ~Google Maps Polyline~, -> String  
    }  
}
```

```

    stops: [ ~Cities~ ],
  }
}
{
  command: "addRoute",
  code: ~Response Code~,
  data: {
    id: ~Route ID~
  }
}

```

```
// 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~,
    password: ~User Password~
}
Route {
    id: ~Route ID~,
    info: ~Short Route Info~,
    polyline: ~Google Maps Polyline~,
    stops: [ ~Cities~ ]
    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;
}

```

-> String

-> String

-> Int

-> RouteInfo

-> String

-> City[]

// * _ *

-> City

-> City

-> Double

-> String

-> String

-> Location

```

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:

// TODO

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