# Test – Developer

The goal is to implement a REST- Interface using C# and .NET. The following requirements have to be met:

- It should be possible to manage people and their favorite colors over the interface
- The data should be read from a CSV file without the need to customize the CSV itself
- All people with specific favorite color can be determined over the interface

Example for the CSV file:

Müller, Hans, 67742 Lauterecken, 1

Petersen, Peter, 18439 Stralsund, 2

Johnson, Johnny, 88888 made up, 3

Millenium, Milly, 77777 made up too, 4 Müller, Jonas, 32323 Hansstadt, 5 Fujitsu, Tastatur, 42342 Japan, 6 Andersson, Anders, 32132 Schweden - Bonus, 2 Bart, Bertram, 12313 Wasweißich, 1 Gerber, Gerda, 76535 Woanders, 3 Klaussen, Klaus, 43246 Hierach, 2

The numbers of the fourth column should match the following colors:

| | |
|---|---|
| 1 | blue |
| 2 | green |
| 3 | purple |
| 4 | red |
| 5 | yellow |
| 6 | turquois |
| 7 | white |

The format to return the data should match the following (Content-Type: application/json):

The implemented REST- Interface should include the following methods:

GET /persons

Methods to get all available users from the CSV

Return Example:

```
[
{
"id" : 1,
"name" : "Hans",
"lastname": "Müller",
"zipcode" : "67742",
"city" : "Lauterecken",
"color" : "blue"
},
{
```

```
"id" : 2,
...
}
]
```

### GET /persons/{id}

Method to get the person with the given ID from the CSV file.

Since the file does not include an ID, the appropriate line number is relevant (first line – line 0).

Example: id 1 - Petersen, Peter, 18439 Stralsund, blue ; in case there is no person with this exact id, a suitable Http status code needs to be used.

Return example:

```
{
"id" : 1,
"name" : "Hans",
"lastname": "Müller",
"zipcode" : "67742",
"city" : "Lauterecken",
"color" : "blue"
}
```

### GET /persons/color/{color}

Method to get all people with the same favorite color. The color accordingly needs to be coded.

Return example:

```
[
{
"id" : 1,
"name" : "Hans",
"lastname": "Müller",
"zipcode" : "67742",
"city" : "Lauterecken",
"color" : "blue"
},
{
"id" : 2,
…

}

]
```

Tasks:

1. Read the CSV file and cache in a Json fitting model class (needs to be implemented). Ideally this should happen in a class, that abstracts the access to the file, thus this could be replaced through a simple database.
2. Implement the given REST- interface. The interface accesses the persistence class via Dependency Injection.
3. Write fitting unit tests for the interface (e.g. GetUsersWithColorTest(), GetAllUsersTest())
4. Bonus:
    - Implementation as a MSBuild project for continuous integration on TFS
    - Implement an additional method POST /persons which adds a new record to the CSV file
    - Entity Framework- connection to a SQL database to persist the data