

Simple specification

In the attached ZIP archive you will find C# .NET 4.8 console project. There you will find console app named `partycli.exe` that shows and saves servers received from API:

Currently this console application has following functions:

- This should fetch servers from API, store them in persistent data store and display each server (name, load, status) and total number of servers in the console:

Unset

```
partycli.exe server_list
```

- This should fetch specific country (France) servers from API, store them in persistent data store and display each server (name, load, status) and total number of servers in the console:

Unset

```
partycli.exe server_list --france
```

- This should fetch specific TCP protocol servers from API, store them in persistent data store and display each server (name, load, status) and total number of servers in the console:

Unset

```
partycli.exe server_list --TCP
```

- This should fetch servers from persistent data store and display each server (name, load, status) and total number of servers in the console:

Unset

```
partycli.exe server_list --local
```

`partycli.exe` for now it's simple console app and written without having in mind that it could grow in the near future into enterprise grade cli monster:

1. There might be more parameters for the app.
 2. Persistent data store provider/storage type/libraries might change.
 3. Servers might be displayed differently in the console or even displayed with colors.
 4. Different API might be choosen.
- It should be fairly easy to adapt current app code to the upcoming requirements. So choose your architecture wisely!

Your goal is to improve this code, make it more robust, scalable, maintainable, testable - just easier to work with.

Few simple requirements

- Refactor existing application
- Write high quality, scalable, maintainable, testable code
- Try to follow modern .NET development practices:
- Don't reinvent the wheel! If you find a nice library/framework that can make your life easier use it!
- Have fun!