

# Exercises: SOAP

- Server: Spyne package
- <https://spyne.io>
- Annotation-based
- `@srpc(InputType, OutputType)`
- Automatically produces wsdl file
- Simple/Complex Types

# Exercises: SOAP

- GitHub: [DS\\_Examples/soap/soap\\_server.py](#)
  1. Define a new function in the interface that calculates the n-th member of the Fibonacci sequence and returns the result to the client.
  2. Adapt the SOAP-client ([soap\\_client.py](#)) to call this function with a value given as a command-line parameter

# Exercises: SOAP

3. (Homework) Define a complex type Person with the following data fields:

- Forename (string)
- Surname (string)
- SSN (integer)

Add a function to the SOAP interface taking a Person as a parameter and returning true if the SSN number is even.

**Hint:** Lookup the `ComplexModel` class in the Spyne documentation!

# Exercises: REST

- We will use the Flask framework with the Flask-Restx extension
- <https://flask-restx.readthedocs.io/en/latest/index.html>
- Example: DS\_Examples/rest/rest\_server.py

# Exercises: REST

4. Implement the DELETE operation so that a pet can be deleted from the list.
5. Implement the POST operation so that a new pet can be created. If a pet with the given name already exists, a new pet with the same name and a number appended to it will be created
6. Create a new resource „birds“ that stores birds with the following fields: name, species, color. Implement the GET, POST and DELETE operations.

**Remember:** PUT should be an idempotent operation, while POST is not