Your task:

Implement a simple message board web application. The application should have two services: createMessage and listMessages. The services should be implemented using a protocol (of your choice) running on HTTP.

CreateMessage receives a message in the request and persists it in the application. The message should have 4 fields: title, content, sender and url. Url should be a valid url, the others can be arbitrary strings (with limited lengths).

ListMessages service lists all the messages persisted in the application.

The service should support two response versions within the same endpoint. The caller is able to define which response version he can handle.

- Messages returned by the first version should contain only title, content and sender fields. The first version must not accept any other parameters than the version parameter.
- Messages returned by the second version should return all 4 fields. The second version also takes a parameter which defines the format in which the response is returned (supported formats could be e.g. JSON and XML).

Bonus task (implement this if you have time and everything else is done):

- Use two different protocols, one for each service, e.g. SOAP with one and REST with the other.

Notes:

- The application is going to be a huge success some day, so it should scale well and be easily extendable. Especially, there will be need for more response versions in listMessages service in the future.
- The application is meant to be used as a part of a larger SOA solution, i.e. GUI might be implemented in another application, by just using services provided by the message board application.
- For simplicity the data does not have to be persisted on disk (it is OK to lose all data when the server is shutdown). However, the design should be such that a real database could be taken into use easily.

Building & deploying:

- The solution should be returned in one self-contained zip file. It should contain the source code and everything needed for running the application from the command line - including, but not restricted to: configuration files, scripts and instructions. You can assume most common command line tools to be available.
- The script should start a web server that will run the application.

Restrictions:

- The application MUST NOT require any external database server!
- There MUST NOT be any dependencies to any specific OS!