



Welcome to CraftingSoftware coding challenge for Erlang Engineer!

In this coding challenge we kindly ask you to implement a rather contrived HTTP job processing service.

A job is a collection of tasks, where each task has a name and a shell command. Tasks may depend on other tasks and require that those are executed beforehand. The service takes care of sorting the tasks to create a proper execution order.

An example request body:

```
{
  "tasks": [
    {
      "name": "task-1",
      "command": "touch /tmp/file1"
    },
    {
      "name": "task-2",
      "command": "cat /tmp/file1",
      "requires": [
        "task-3"
      ]
    },
    {
      "name": "task-3",
      "command": "echo 'Hello World!' > /tmp/file1",
      "requires": [
        "task-1"
      ]
    },
    {
      "name": "task-4",
      "command": "rm /tmp/file1",
      "requires": [
        "task-2",
        "task-3"
      ]
    }
  ]
}
```

For which an example response might look like the following:

```
{
  "tasks": [
    {
      "name": "task-1",
      "command": "touch /tmp/file1"
    },
    {
      "name": "task-3",
      "command": "echo 'Hello World!' > /tmp/file1"
    },
    {
      "name": "task-2",
      "command": "cat /tmp/file1"
    },
    {
      "name": "task-4",
      "command": "rm /tmp/file1"
    }
  ]
}
```

Additionally, the service should be able to return a bash script representation directly:

```
#!/usr/bin/env bash
touch /tmp/file1
echo "Hello World!" > /tmp/file1
cat /tmp/file1
rm /tmp/file1
```

Please include any instructions you deem necessary to be able to test the execution of the solution.