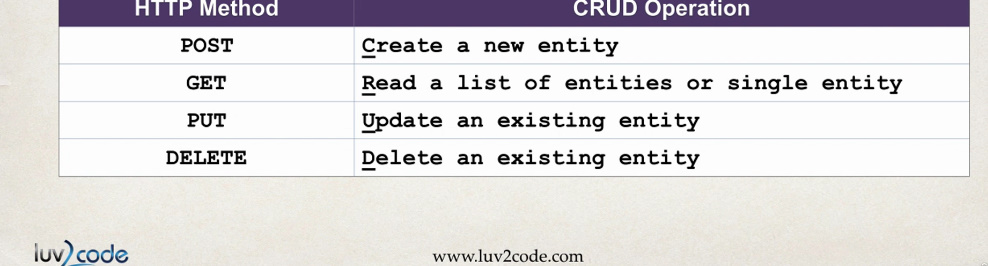
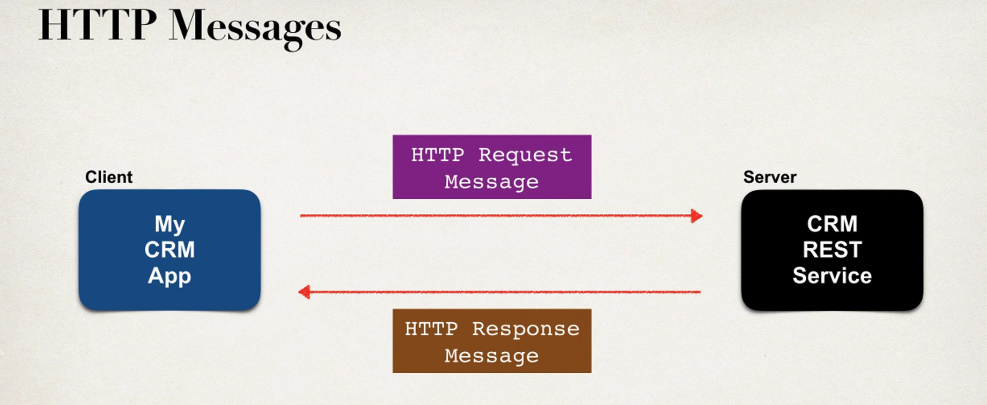
**REST prin HTTP**

* Cel mai comun mod de a folosi REST e prin HTTP
* Putem folosi HTTP methods pentru CRUD operatii:

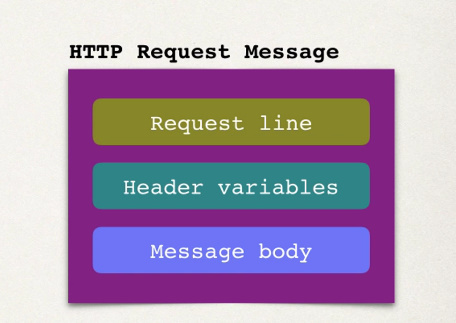


* Mod de functionare:



* HTTP Request Message e cel care face requestul, si i se ofera apoi un raspuns.

**Un HTTP Request Message are structura:**



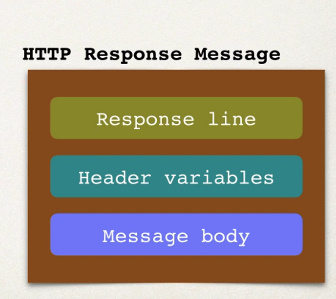
Request line – HTTP command(get, put, post, delete)

Header variables – request metadata, inf aditionale

Message body – continutul mesajului propriu zis

**HTTP Response Message**

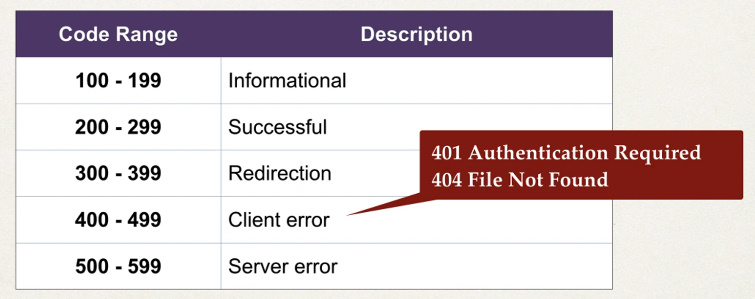
Este raspunsul care vine pentru request.



Response line – server protocol and status code(un HTTP status code ca 201, 215 etc)

Header variables – response metada, info despre data, ca XML, JSON etc/

Message body – continutul mesajului, ca lista propriu zisa de date

**Status Codes**

**MIME Content Types**

* MIME – Multipurpose Internet Mail-Extension
* El doar descrie continutul mesajului returnat
* Syntax: type/sub-type
* Ex:

text/html – dam text la browser ca sa fie citit ca HTML

text/plain – text ce sa fie citit in intregime cum este el

application/json – datele in format JSON sunt trimise la applicatie

application/xml

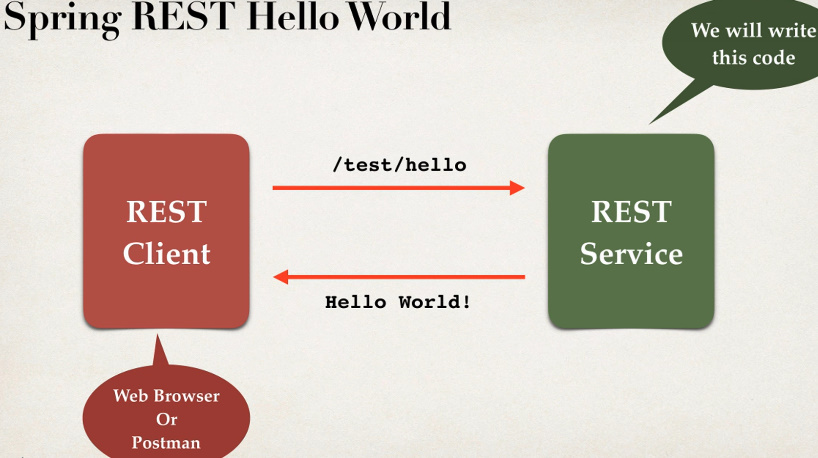
* Aceasta e informatie returnata pentru client

**Client Tool**

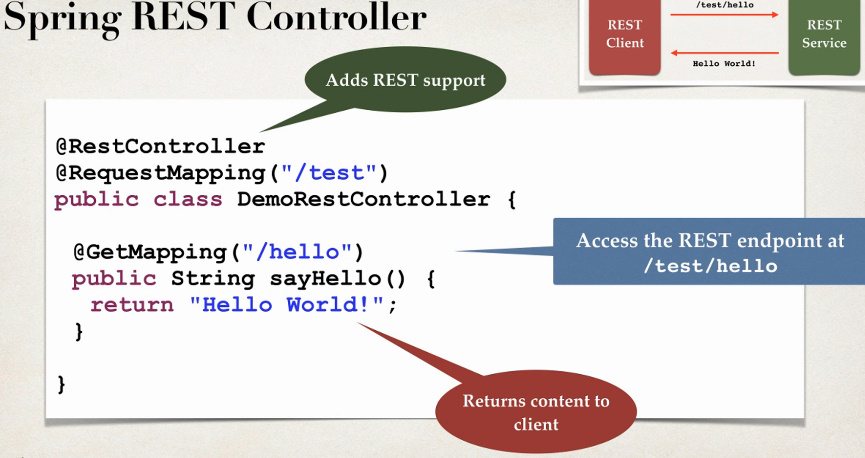
* Il folosim pentru a trimite HTTP requests la REST Web Service / API
* Tools: curl, **Postman** etc.

**Create Spring REST Controller**

* Pentru a crea un REST Controller, folosim anotatia @RestController, care e o extensie de la @Controller cu @RequestMapping
* @RestController se ocupa de requests/responses
* Spring REST converteste automat Java POJOs in JSON cu Jackson



* **Ex:**



* Putem testa si cu Web Browser si Postman.
* Totusi Postman, e pentru advanced Requests, caci browser e mai mult pentru get