

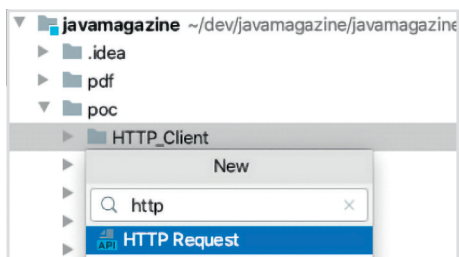
# HTTP Client in IntelliJ IDEA

Het bedrijf JetBrains heeft voor vele talen een populaire Integrated Development Environment (IDE) gemaakt. Denk aan IntelliJ IDEA, PyCharm en onder vele andere WebStorm. Voor de betaalde (Ultimate of Professional) versies van deze IDE's is een handige plugin beschikbaar die standaard mee-ge-bundled is in die versies. De HTTP Client. Een plugin voor het maken van HTTP requests en nog veel meer, zoals beschreven zal worden in dit artikel.

De HTTP Client is helemaal op tekst gebaseerd is daardoor simpel in het gebruik en makkelijk te leren.

## Getting started...

Om te beginnen hebben je een http bestand (extensie .http) nodig dat je in elk project kan plaatsen of als scratch bestand kan maken. Bijvoorbeeld: my-requests.http



Afbeelding 1.

Nu is er een tekst bestand gemaakt waar je je requests in kan plaatsen. Requests worden simpel gescheiden met drie hashes (###) op een nieuwe regel. Een enkele hash kan gebruikt worden voor commentaar. Execute het request met de groene 'play' button voor de GET of met de gekijkte sneltoetsen van de IDE. Als je de GET uitvoert dan krijg je de output te zien, maar het wordt ook opgeslagen in de <project>/idea/httpRequests folder. In die folder wordt ook een http-request-log en een cookie log bijgehouden. Deze is ook te openen via het menu.

Op elk request kunnen annotaties geplaatst worden om bepaalde acties te voorkomen of juist af te dwingen.

```
# @no-log: er wordt geen log bijgehouden
# @no-redirect: het request mag geen redirects volgen
# @no-cookie-jar: bewaar het cookie niet
# @use-os-credentials: gebruik de OS waarde credentials bij je request.
```

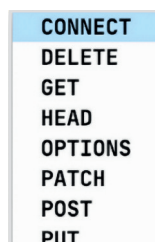
Alle standaard HTTP request worden ondersteund (afbeelding 2) en daar kan je leuke dingen mee doen. Zo kan je bijvoorbeeld testen uitvoeren op het request (zie listing 1). De testen worden in JavaScript geschreven. Op alles in het bestand krijg je "code completion", zoals je gewend bent binnen IntelliJ. Om het leven nog makkelijk te maken wordt een client en een response object gecreëerd. Op afbeelding 3 kan je zien dat de testen geslaagd zijn en op afbeelding 4 kan je zien wat er getest is en dus ook wat er eventueel is fout gegaan. In listing 2 zie je ook {{host}} en {{json}} staan in plaats van de echte host. Dit zijn variabelen.

## Variabelen

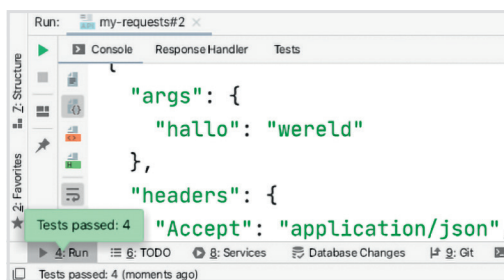
De HTTP Client kan werken met variabelen. Deze kunnen uit een Regulier of een Private bestand komen. Deze bestanden kan je eventueel ook zelf maken: http-client.env.json / http-client.private.env.json.



**Ivo Woltring** is Software Architect en Codesmith bij Ordina JTech en houdt zich graag bezig met nieuwe ontwikkelingen in de software wereld.

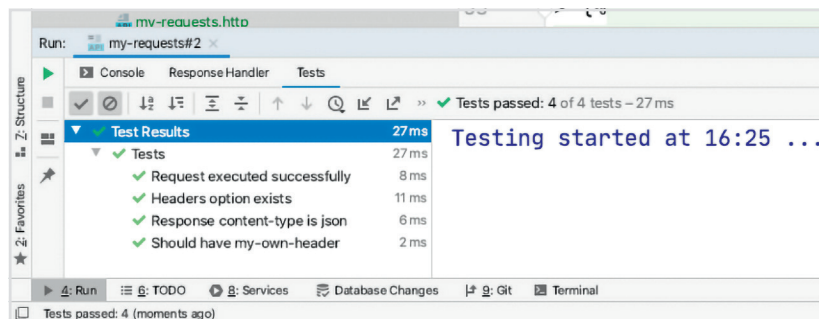


Afbeelding 2.



Afbeelding 3.

Afbeelding 4.



In listing 2 staat een voorbeeld van zo'n environment bestand. Je kan er allerlei soorten informatie in kwijt.

```
"httpbin": {
  "host": "httpbin.org",
  "json": "application/json"
},
"api.github.com": {
  "host": "api.github.com"
}
}
```

Listing 2.

Het private environment bestand wordt standaard niet aan het VCS (Version Control System) toegevoegd door de IDE en daar worden meestal authenticatie tokens en wachtwoorden in bewaard. Het groeperen en delen van je requests kunnen zo een stuk makkelijker gemaakt worden. Buiten deze variabelen zijn er ook een paar zogenaamde “Dynamic variables” beschikbaar en dat zijn de \$uuid, \$timestamp en \$randomInt. Deze kunnen op elk moment gebruikt worden en zullen respectievelijk een uuid, een unix timestamp en een integer genereren dat tussen de 0 en de 1000 ligt. Zie listing 3 voor een voorbeeld en listing 4 voor een stukje van het response. Heel handig voor testen.

```
###
POST https://{{host}}/post
Accept: {{json}}
Content-Type: {{json}}

{
  "uuid": "{{${uuid}}}",
  "timestamp": "{{${timestamp}}}",
  "randomInt": "{{${randomInt}}}"
}
```

Listing 3.

```
[...]
"json": {
  "randomInt": "779",
  "timestamp": "1594997737",
  "uuid": "f6d04da0-eeefc-4aa0-ae1a-c6a2b38c9844"
},
[...]
```

Listing 4.

Naast deze variabelen is het ook nog mogelijk om variabelen “at runtime” te bewaren via de `client.global.set("variabele_naam", variabele_value)` in een test blok die dan in een volgend request weer gebruikt kan worden met `{{variabele_naam}}`.

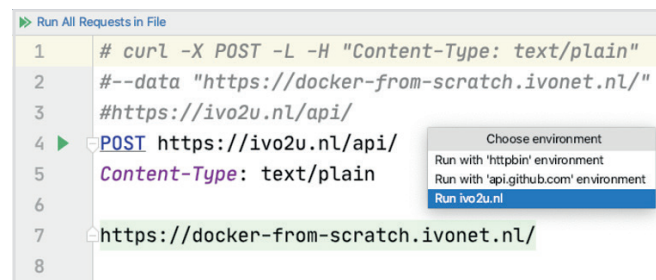
## cURL

Nog een handige feature van de HTTP Client is dat het heel makkelijk is een curl commando kan converteren naar het ondertussen

```
### Check response status, headers, content-type and my-own-header
GET https://{{host}}/get?hallo=wereld
Accept: {{json}}
my-own-header: Hello, world!

> {%
  client.test("Request executed successfully", function() {
    client.assert(response.status === 200, "Response status is not 200");
  });
  client.test("Headers option exists", function() {
    client.assert(response.body.hasOwnProperty("headers"), "Cannot find 'headers' option in response");
  });
  client.test("Response content-type is json", function() {
    var type = response.contentType.mimeType;
    client.assert(type === "application/json", "Expected 'application/json' but received '" + type + "'");
  });
  client.test("Should have my-own-header", function() {
    client.assert(response.body.headers["My-Own-Header"] === "Hello, world!", "Bye bye world.");
  });
}%
```

Listing 1.



Afbeelding 5.

bekende tekstformaat. Dit kan heel handig zijn als je bijvoorbeeld Postman requests wilt porten naar de HTTP Client of als je eerst wat hebt uitgetest in de terminal. Als je al een curl commando in je clipboard hebt gekopieerd zal deze automatisch geplakt worden in het tekstveld. In afbeelding 5 kan je zien wat het resultaat is van de conversie en welke opties je krijgt bij het draaien. In dit voorbeeld is uitgegaan van het environment bestand van listing 3.

## Conclusie

Voor ontwikkelaars die veel gebruik maken van hun commerciële versie van IntelliJ, of andere IDE van JetBrains, is de tekst based HTTP Client een super handig tool. Wellicht zo handig dat geen andere tools meer nodig zijn om dit soort requests te doen. ■

## REFERENCES

<https://www.jetbrains.com/> (de makers 😊)  
<https://httpbin.org/> (test server)  
<http://ivo2u.nl/oz> (voorbeeld code)