HTTP ET ANATOMIE D'UNE APPLICATION

Yann Rotella

Université de Versailles Saint Quentin en Yvelines
Université Paris-Saclay

8 avril 2021

HyperText Transfer Protocol

- Protocole pour servir les documents Web.
- Requètes
- Réponses
- Sans état

HTTP 1.1 REQUÈTES

```
GET / HTTP/1.1
Host: www.google.fr

POST /document.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Cookie: sessionid=aa03x;
Content-Length: 10

1234567890
```

HTTP 1.1 RÉPONSES

```
HTTP/1.1 200 OK
Date: Wen, 28 Mar 2020 06:03:00 GMT
Server:
Expires:
Last-Modified:
Set-Cookie: sessionid=
```

HTTP 1.1 RÉPONSES

```
HTTP/1.1 200 OK
Date: Wen, 28 Mar 2020 06:03:00 GMT
Server:
Expires:
Last-Modified:
Set-Cookie: sessionid=
```

Erreurs :

- 1xx : Infos
- 2xx : Ressource trouvée et réponse en fonction
- 3xx : Redirection
- 4xx : Erreurs côté client (404 not found)
- 5xx : Erreurs côté serveur

HTTP 1.1 RÉPONSES

```
HTTP/1.1 200 OK
Date: Wen, 28 Mar 2020 06:03:00 GMT
Server:
Expires:
Last-Modified:
Set-Cookie: sessionid=
```

Erreurs :

- 1xx : Infos
- 2xx : Ressource trouvée et réponse en fonction
- 3xx : Redirection
- 4xx : Erreurs côté client (404 not found)
- 5xx : Erreurs côté serveur

DevTools -> Network !

HTTP 2

- Binaire
- Requètes et réponses parallèles multiplexées
- Temps de latence
- Server Push
- Priorisation des flux
- Compression des en-têtes (HPACK)...

HTTP 3

- Encore peu utilisé

HTTP 3

- Encore peu utilisé
- Une autre histoire...

Navigateurs

- Du point de veu du client, c'est lui qui gère le HTTP
- Affichage du contenu (HTML, CSS)
- Exécution de scripts !
- Plugins (lecture PDF par exemple)

HAMECONNAGE

- `http://cas.uvsq.fr:blabla` : message d'erreur
- Encodage de caractères différents
- Aujourd'hui : Certificats (on verra plus tard) !

Noms de domaine

- `exemple.fr` correspond à un regroupement d'adresses IP
- `.fr` pays, territoire mais aussi d'autres regroupements (Internet Corporation for Assigned Names and Numbers)
- Association franąise pour le nommage internet en coopération (AFNIC)
- Choix du nom de domaine et de l'extension

HÉBERGEMENT ET DEV

- AWS, Heroku
- OVH Start 10M, et bien d'autres
- Glitch et bien d'autres

L'ANTIQUITÉ

- Documents statiques
- Client envoie `GET .../index.html`
- Serveur renvoie `HTML 1.1 200 OK`

L'ANTIQUITÉ

- Documents statiques
- Client envoie `GET .../index.html`
- Serveur renvoie `HTML 1.1 200 OK`

Utilité : pages perso, site informatif, cours

Pages Dynamiques 1.0

La création du document se réalise au moment où la requète est reçue. Le serveur réalise donc :

- des compilations de documents
- des interactions avec d'autres serveurs
- accès à la base de données

Web 2.0, applis web

- Les URLs ne correspondent plus à des fichiers sur le serveur
- Les URLs sont des pointeurs et corespondent à une action
- l'action est reálisée par l'application
- Node.js, Django, Ruby on Rails, Symfony, Zend...

Frameworks backend

- HTTP API (1.1 et 2.0) : analyse et écriture des requètes
- Routeur : définir l'action à réalisée par URL
- Template : génération de pages HTML
- Stockage court : sessions (persistence)
- Base de donées
- Sécurité (Helmet pour Express par exemple)

Express et Node

```
var express = require('express');
var app = express();

app.get('/', ...);
app.post('/form', ...);
app.all('/others', ...);

app.listen(443);
```

ROUTEURS

```
app.get('/url', function(req, res) {
        ...
});
```

ROUTEURS

```
app.get('/url', function(req, res) {
        ...
});

app.get('/url/:a1/:a2', function(req, res) {
        console.log(req.params.a1);
        console.log(req.params.a2);
        ...
```

GÉRER LES REQUÈTES

```
var bodyParser = require("body-parser");
var cookieParser = require("cookie-parser");
```

GÉRER LES REQUÈTES

```
var bodyParser = require("body-parser");
var cookieParser = require("cookie-parser");

app.use(bodyParser.urlencoded
app.use(cookieParser...
```

GÉRER LES REQUÈTES

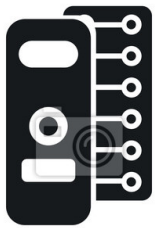
```
var bodyParser = require("body-parser");
var cookieParser = require("cookie-parser");

app.use(bodyParser.urlencoded
app.use(cookieParser...
```

Accès avec `req.query`, `req.body`, `req.cookies`, `req.headers`

REPONSE

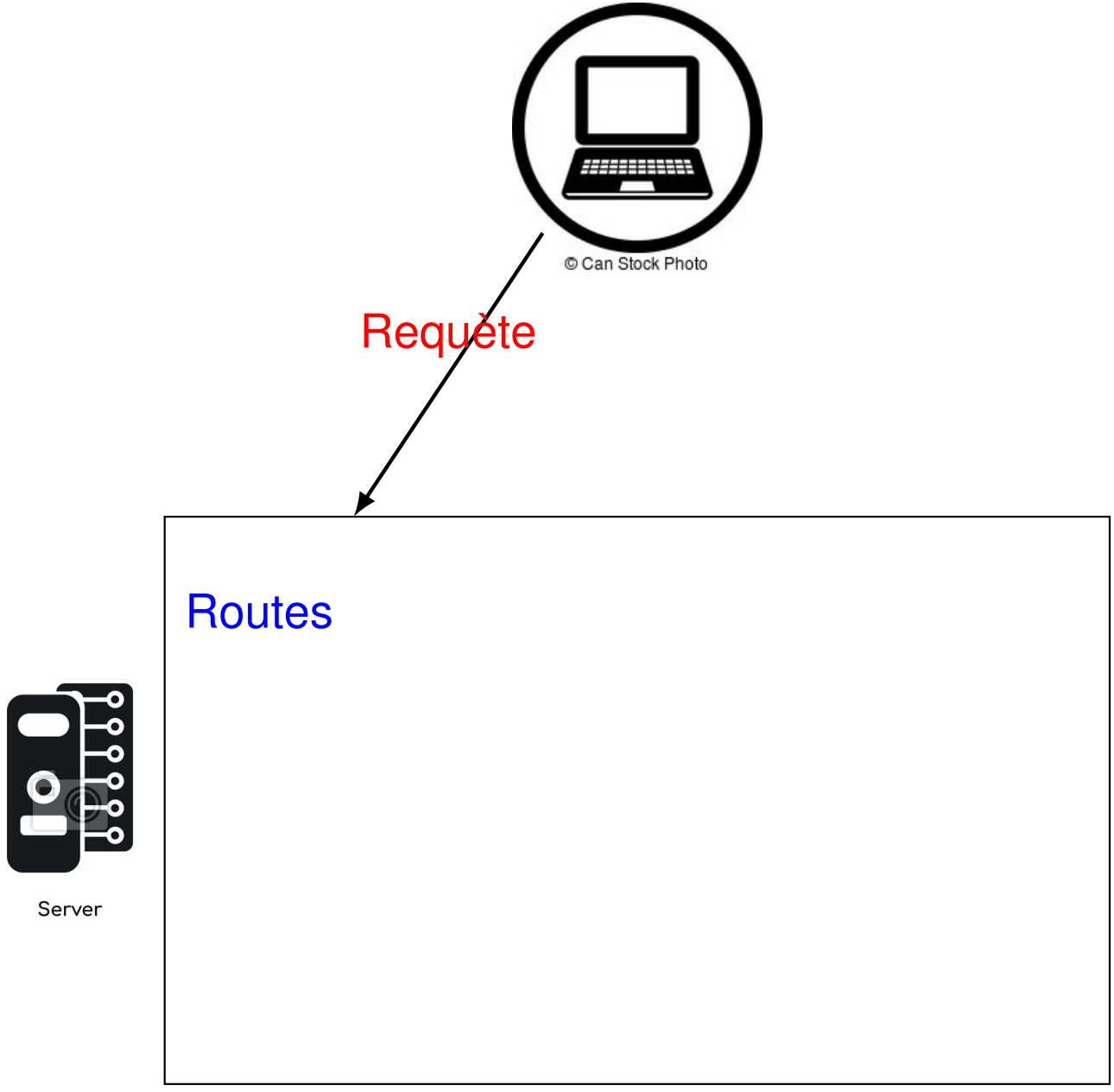
- `res.send` : réponse
- `res.set(field [, value])` : headers of HTTP
- `res.status(404).send('Not Found')`
- `res.sendFile(...), res.download(...)`
- `res.redirect('/other/path');`
- `res.json(...)` : plus tard

FORMAT D'UNE APPLICATION

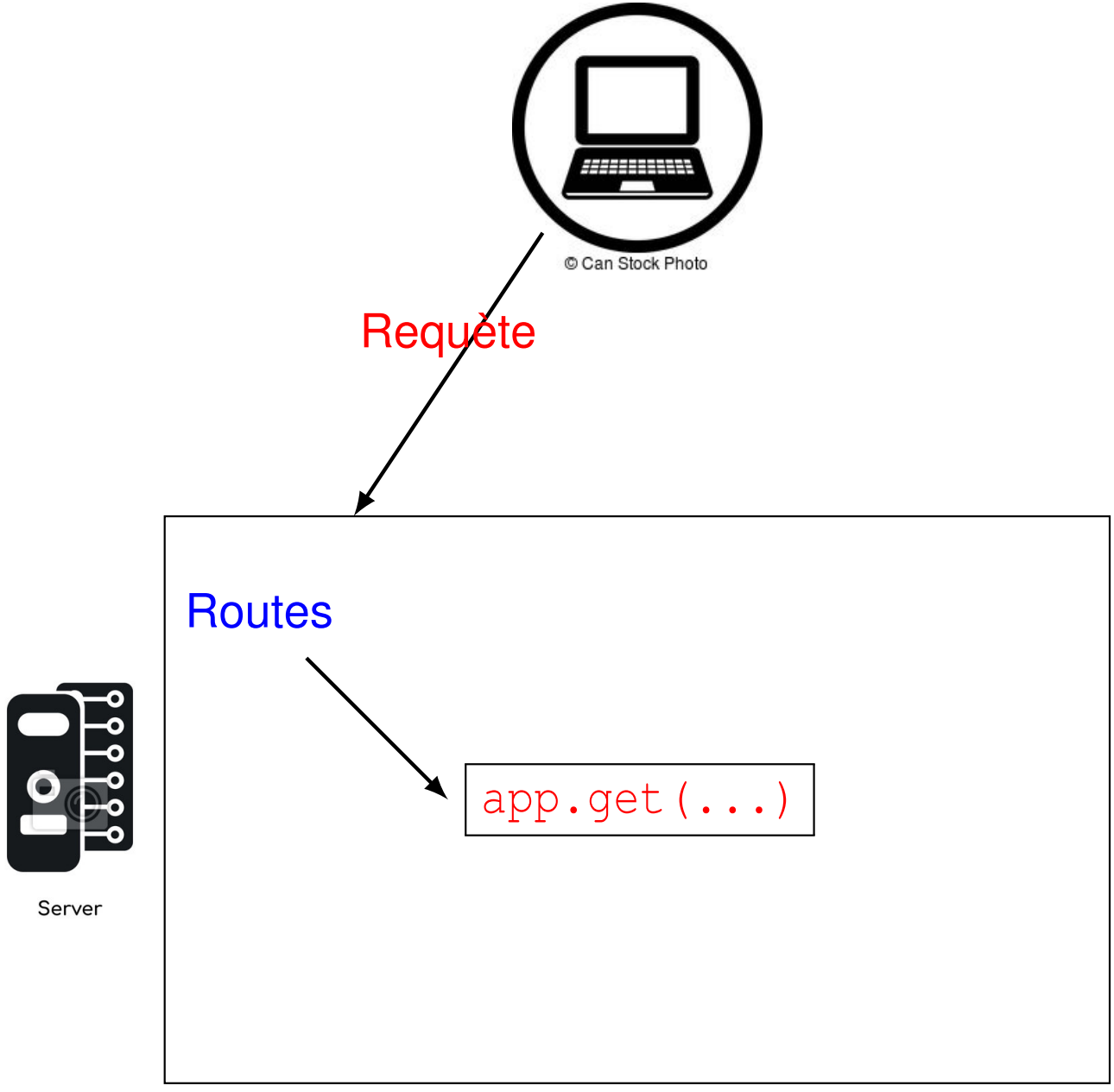

Server

Routes

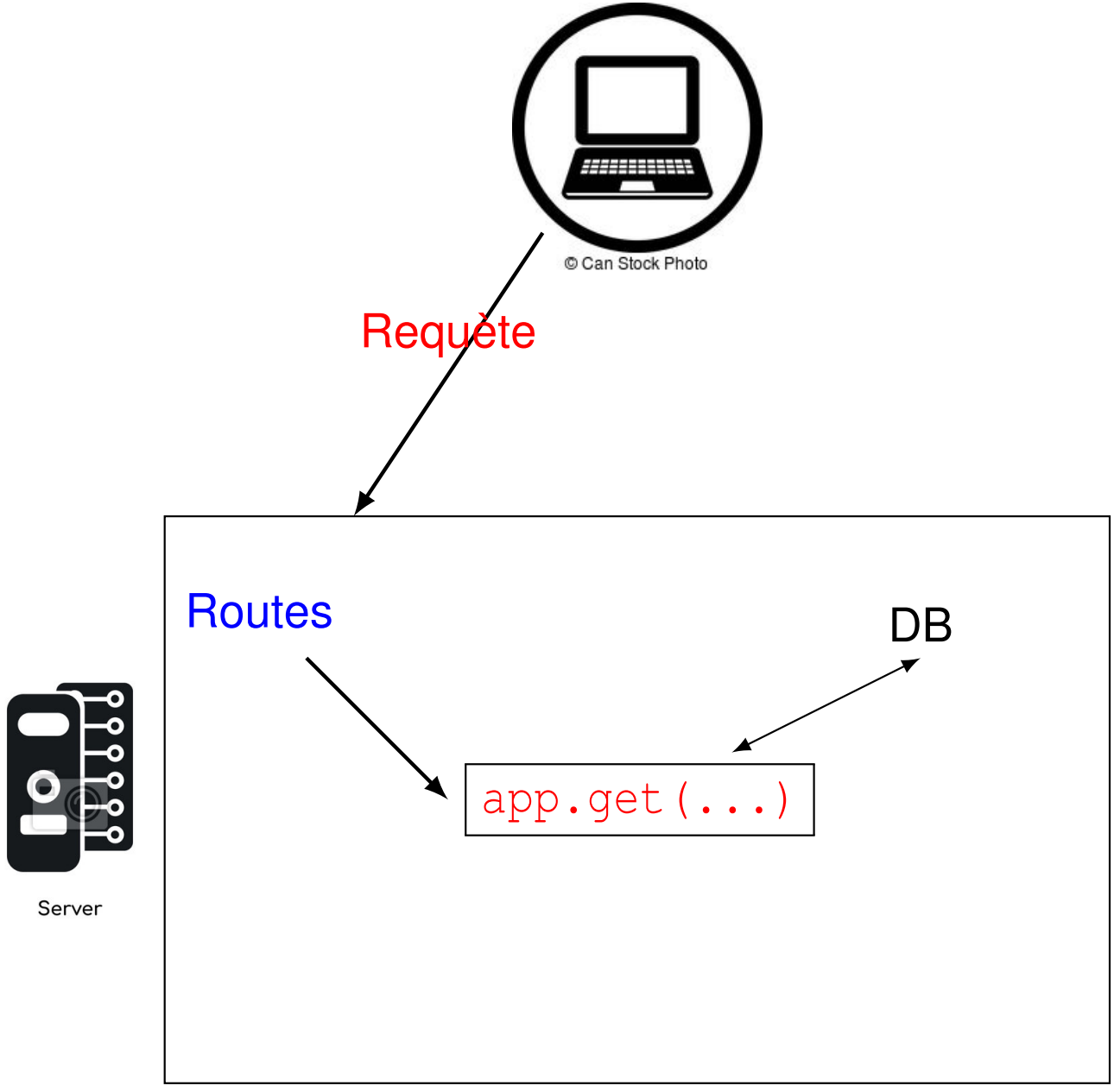
FORMAT D'UNE APPLICATION



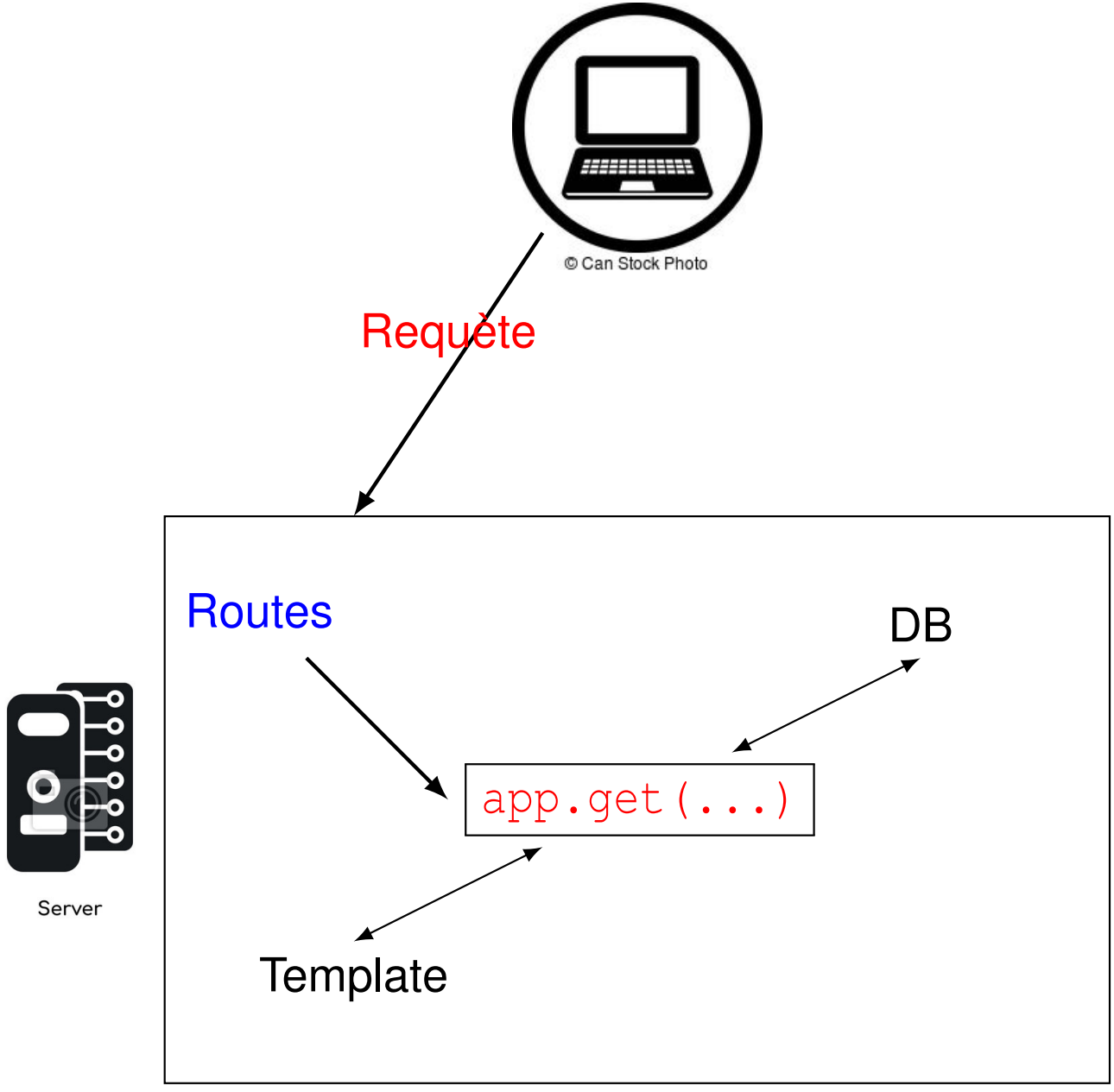
FORMAT D'UNE APPLICATION



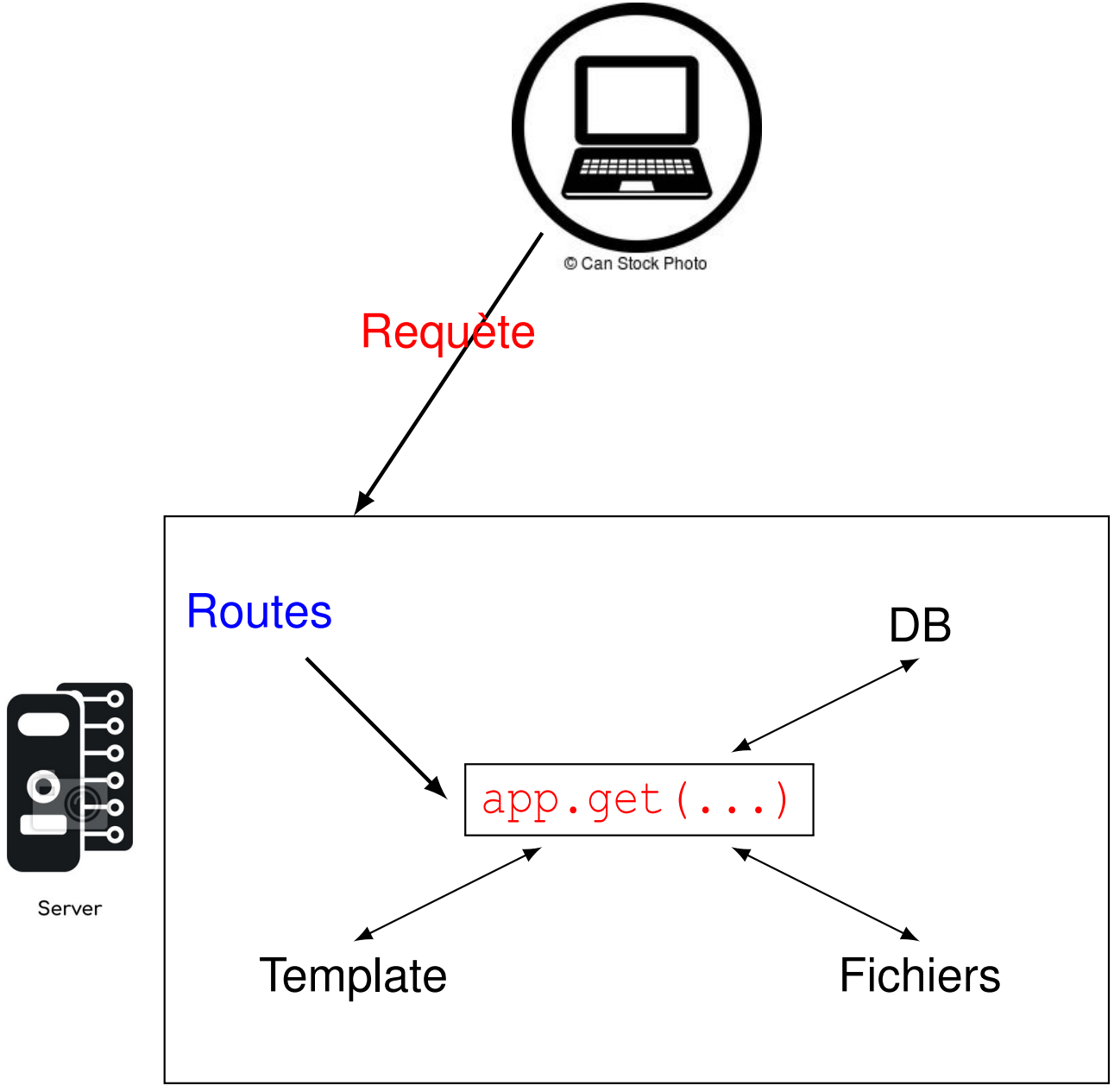
FORMAT D'UNE APPLICATION



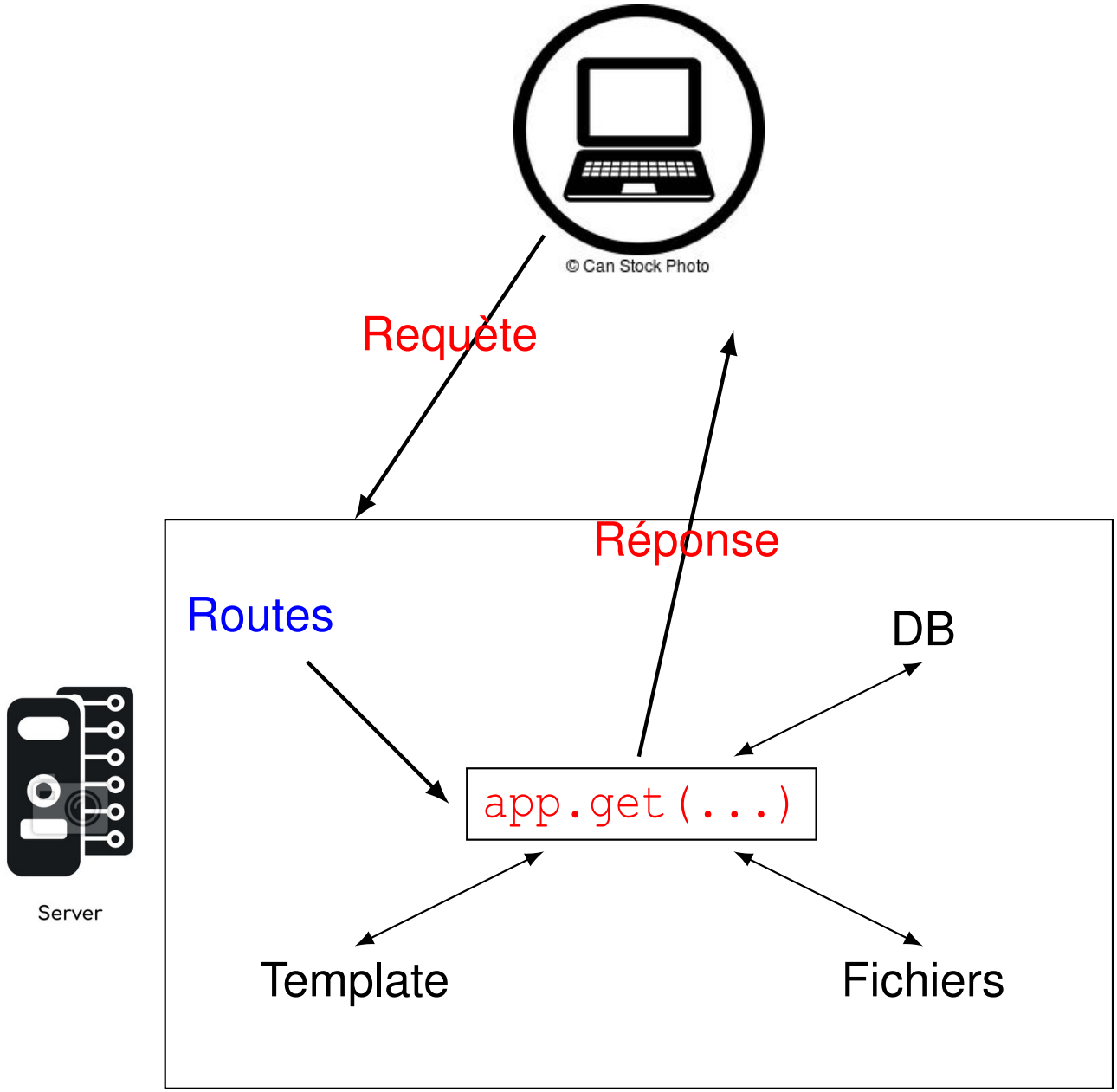
FORMAT D'UNE APPLICATION



FORMAT D'UNE APPLICATION



FORMAT D'UNE APPLICATION



Modèle Vue Controleur

On sépare son application selon une logique.

- Modèle : données à afficher
- Vue : présentation de l'interface graphique
- Controleur : logique des actions