



PolyHacks - Hackatown 2022

GUIDE DU HACKER DÉBUTANT

BEGINNER HACKER'S GUIDE

Salut les hackers!

Bienvenue à l'édition 2022 du hackathon annuel de PolyHx. Cette année, en plus de nos ateliers, nous avons préparé ce petit guide pour vous aider à bâtir votre projet lors de l'évènement. Ce guide vous donne une introduction sur différents sujets comme le développement web, les applications mobiles, les API, les règles de base de gestion des fichiers (Git) et les autres outils de productivité.

L'équipe de PolyHx



Hello hackers!

Welcome to the 2022 edition of PolyHx's annual hackathon. This year, in addition to our workshops, we have prepared this short guide to help you build your project during the event. This guide gives you an introduction to various topics such as web development, mobile applications, APIs, file management basics (Git) and other productivity tools.

PolyHx team



Table des matières - Table of contents

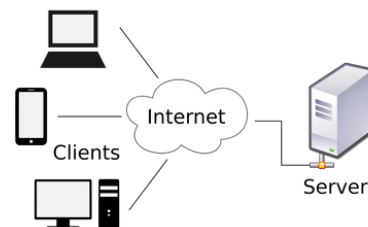
1. Application Web - Web Application	1
a. Front End	2
b. Back End	4
c. Application Bureau - Desktop Application	6
d. Déploiement - Deployment	7
2. Application Mobile - Mobile Application	8
3. APIs	11
4. Git, Github & GitLab	13

1. Application Web

Web Application

Une application web est une application qui est utilisée en ligne à travers un navigateur. Ces applications peuvent être de simples sites web, des applications de messagerie instantanée, etc. Elles sont accessibles à travers les navigateurs car elles sont hébergées sur des serveurs qui sont ensuite déployés et accessibles aux utilisateurs.

A web application is an application that can be used online through a browser. These applications can be simple websites, instant messaging applications, etc. They are accessible through browsers because they are hosted on servers which are then deployed and accessible to users.



Afin d'être en mesure de développer ce type d'application il faut premièrement comprendre l'architecture client-serveur. Essentiellement, elle est basée sur une communication entre le client et le serveur pour faire fonctionner l'application. Pour en savoir, consulter le lien suivant :

In order to be able to develop this type of application, you must first understand the client-server architecture. Essentially, it is based on communication between the client and the server to operate the application. To find out more, consult the following link :

- <https://fr.wikipedia.org/wiki/Client-serveur>

Ces applications fonctionnent ainsi avec deux composantes principales qui sont le front end (côté client) et le back end (côté serveur). Le front end s'occupe de l'apparence de l'application et de la logique afin de permettre l'interaction de l'utilisateur avec les différentes composantes de la vue. Le back end permet de gérer l'application et les demandes du côté du client.

These applications thus work with two main components being the front end (client side) et le back end (server side). The front end takes care of the view of the application and the logic in order to allow the interaction of the user with the different components of the view. The back end makes it possible to manage the application and the requests on the client side.

Front End

HTML & CSS

Les deux technologies incontournables de la création d'une application Web sont HTML et CSS, ce sont des langages qui vont vous permettre de définir le contenu de votre page et de modifier l'apparence des différents éléments de vos pages. Pour débiter, nous vous invitons à consulter :

- https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web

Javascript

Par la suite, pour rendre vos pages plus interactives, vous pouvez ajouter des scripts JavaScript. Ceux-ci vous permettront par exemple de manipuler les éléments de votre page, de communiquer avec des services externes (voir la section API) et pleins d'autres choses. Le site de Mozilla présente aussi un guide sur JavaScript :

- https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps.

Une très bonne alternative est le site

The two essential technologies for creating a Web application are HTML and CSS, these are languages that will allow you to define the content of your page and modify the appearance of the different elements of your pages. To get started, we invite you to consult :

Later, to make your pages more interactive, you can add JavaScript scripts. These will allow you, for example, to manipulate the elements of your page, to communicate with external services (see the API section) and many other things. The Mozilla site also has a JavaScript guide :

A very good alternative is the site

- <https://javascript.info>

Frameworks Frontend

Si vous souhaitez mettre en place des interfaces plus complètes et complexes, nous vous suggérons d'utiliser un Framework Frontend comme ReactJS, Angular, VueJS ou d'autres. L'utilisation d'un Framework peut sauver beaucoup de temps puisqu'il met à votre disposition des éléments déjà prêts à utiliser. Le site de Mozilla a aussi une page dédiée aux Frameworks Frontend.

If you want to implement more complete and complex interfaces, we suggest using a Frontend Framework like ReactJS, Angular, VueJS or others. Using a Framework can save a lot of time since it provides you with elements that are already ready to use. The Mozilla site also has a page dedicated to Frontend Frameworks.

- [https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side JavaScript frameworks/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction)

VueJS

- [https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side JavaScript frameworks/Vue getting started](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Vue_getting_started)
- <https://vuejs.org/>

ReactJS

- [https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side JavaScript frameworks/React getting started](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started)
- <https://reactjs.org/>

Angular

- [https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side JavaScript frameworks/Angular getting started](https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Angular_getting_started)
- <https://angular.io>

Atelier de PolyHx sur Angular - PolyHx's Angular Workshop

<https://www.youtube.com/watch?v=4uNYRR7evxc>

Back End

Le backend est considéré comme le moteur des applications web, la partie cachée de l'iceberg. Il est principalement constitué d'un serveur qui s'occupe de recevoir, de traiter et de répondre aux demandes du client. Il est souvent accompagné d'une base de données qui contient des informations que le serveur peut transmettre au client sur demande de ce dernier.

The backend is considered the engine of web applications, the hidden part of the iceberg. It mainly consists of a server that takes care of receiving, processing and responding to client requests. It is often accompanied by a database that contains information that the server can transmit to the client on request of the latter.

- <https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/>

Langages de programmation - Coding languages

Il existe plusieurs langages permettant de faire le développement côté serveur. Néanmoins les langages de programmation principaux sont javascript, python, C# et java.

There are several languages available for server-side development. Nevertheless the main programming languages are javascript, python, C# and java.

Pour en savoir plus sur d'autres langages, vous pouvez suivre le lien suivant :

To learn more about other languages, you can follow the following link :

- <https://www.geeksforgeeks.org/top-7-programming-languages-for-backend-web-development/>

Nous vous recommandons de consulter les documents suivantes.

We recommend that you consult the following documents.

- Python : <https://www.python.org/doc/>
- C# : <https://docs.microsoft.com/en-us/dotnet/csharp/>
- Javascript : <https://devdocs.io/javascript/>
- Java (script) : <https://spring.io/>

Frameworks Backend

Il existe plusieurs frameworks basés sur différents langages qui fournissent des outils importants afin de faciliter le développement de serveurs fiables et robustes.

Vous pouvez consulter cette ressource pour découvrir les principaux frameworks utilisés :

There are several frameworks based on different languages that provide important tools to facilitate the development of reliable and robust servers.

You can consult this resource to discover the main frameworks used :

- <https://www.monocubed.com/best-backend-frameworks/>

Voici une liste plus réduite de quelques frameworks très populaires que nous vous suggérons d'utiliser.

Here is a smaller list of some very popular frameworks that we suggest you use.

Django

Django est un framework de développement web en python, il est très pratique car il permet un développement rapide tout en vous assurant de développer une application sûre et fiable.

Django is a web development framework in python, it is very practical because it allows rapid development while ensuring that you develop a safe and reliable application.

Nous vous invitons à consulter la documentation de Django à l'adresse suivante :

We invite you to consult the Django documentation at the following address :

- <https://www.djangoproject.com/>

Express JS

C'est un framework de développement web basé sur NodeJS. NodeJS étant un environnement d'exécution pour le développement d'applications réseau en javascript.

It is a web development framework based on NodeJS. NodeJS being a runtime environment for the development of network applications in javascript.

Express fournit des outils importants pour le développement web. Nous vous invitons à consulter le liens en dessous pour plus d'informations sur Express JS et Node JS :

Express provides important tools for web development. We invite you to consult the links below for more information on Express JS and Node JS :

- <https://nodejs.org/en/>
- <http://expressjs.com/>

Application bureau - Desktop Application

Jusqu'ici nous avons parlé d'applications web qui sont accessibles depuis internet à travers un serveur. Il existe aussi des applications bureau qui sont téléchargeables sur vos ordinateurs et qui ne nécessite pas de passer par un navigateur comme chrome pour y accéder. Un exemple serait l'application NotePad sur vos ordinateurs Windows.

So far we have talked about web applications that are accessible from the internet through a server. There are also desktop applications which can be downloaded on your computers and which do not require you to go through a browser like Chrome to access them. An example would be the NotePad application on your Windows computers.

Si vous voulez transformer votre application web en une application bureau, le framework electron vous permettra de le faire. Voici quelques ressources pour vous familiariser avec le framework electron :

- <https://www.electronjs.org/>
- <https://www.twilio.com/blog/an-introduction-to-building-desktop-applications-with-electron>
- <https://medium.com/developers-writing/building-a-desktop-application-with-electron-204203eeb658>

If you want to transform your web application into a desktop application, the electron framework will allow you to do so. Here are some resources to familiarize yourself with the electron framework :

Déploiement - Deployment

Afin de rendre votre application disponible à l'extérieur de votre ordinateur vous pouvez la déployer sur un serveur externe. Voici une liste de ressources utiles :

In order to make your application available outside your computer you can deploy it on an external server. Here is a list of useful resources :

- VueJS : <https://cli.vuejs.org/guide/deployment.html#platform-guides>
- React : <https://blog.logrocket.com/8-ways-to-deploy-a-react-app-for-free/>
- Angular : <https://angular.io/guide/deployment#deploy-to-github-pages>

2. Application mobile

Mobile Application

Le développement mobile est similaire au développement web du point de vue de l'architecture client-serveur. Cependant, ce qui est particulier au développement mobile est le fait qu'on aura besoin d'outils différents pour développer sur les appareils **IOS** ou **Android**.

Vous pouvez lire sur le processus en général ici :

Mobile development is similar to web development from a client-server architecture perspective. However, what is particular to mobile development is the fact that one will need different tools to develop on IOS or Android devices.

You can read about the overall process here :

- <https://aws.amazon.com/mobile/mobile-application-development/>

Front End

Environnement de développement - Development Environment

Voici les environnements de développement qui contiennent un émulateur, un outil qui vous permettra de visionner votre interface durant le développement de votre application.

Here are the development environments that contain an emulator, a tool that will allow you to view your interface during the development of your application.

Android : Angular Studio

- <https://developer.android.com/studio/intro>

IOS : Xcode

- <https://developer.apple.com/documentation/xcode>

Langages de programmation - Programming Languages

Android : Kotlin (recommandé par Google - recommended by Google)

- <https://developer.android.com/courses/android-basics-kotlin/course>

Android: | Java (Android SDK)

- <https://docs.oracle.com/en/middleware/developer-tools/jet/tutorials/jetma/index.html>

IOS: Swift

- <https://docs.swift.org/swift-book/LanguageGuide/TheBasics.html>

Quadriciels - Frameworks

Android & IOS : Flutter

- <https://flutter.dev/learn>

Atelier de Poly Hx sur la programmation mobile en Flutter -
PolyH'x workshop on mobile programming in Flutter

https://www.youtube.com/watch?v=XLSJTmA9Qz4&ab_channel=PolyHx

Back End

Vous pouvez développer le backend de votre application mobile avec les mêmes outils mentionnés dans la section sur le développement Web. Cependant, si vous préférez de vous concentrer uniquement sur la partie client de votre application, il existe des quelque solution back-end déjà prêts pour vous :

You can develop the backend of your mobile app with the same tools mentioned in the web development section. However, if you prefer to focus only on the client part of your application, there are some ready-made back-end solutions for you :

Firebase

- <https://firebase.google.com/docs/guides>
- IOS : <https://firebase.google.com/docs/ios/setup>
- Android : <https://firebase.google.com/docs/android/setup>

ASW

IOS & Android

- <https://docs.aws.amazon.com/wellarchitected/latest/serverless-applications-lens/mobile-backend.html>

3. APIs et données ouvertes

APIs and open data

Les API (Application Programming Interfaces) vous permettent d'interagir avec d'autres applications ou services mis à disposition pour les utilisateurs par les entreprises et les institutions. Nous avons rassemblé quelques ressources qui pourraient vous intéresser.

APIs (Application Programming Interfaces) allow you to interact with other applications or services made available to users by companies and institutions. We've put together a few resources that might interest you.

STM - Société de Transport de Montréal

La Société de Transport de Montréal (STM) met à disposition plusieurs ensembles de données sur le réseau de bus et de métro incluant notamment la position des stations, les horaires, la position en temps réel des bus et métro, etc.

The Société de Transport de Montréal (STM) provides several sets of data on the bus and metro network, including station positions, timetables, real-time bus and metro positions, etc.

Pour utiliser l'API de la STM vous aurez besoin d'une clé de connexion. Pour obtenir celle-ci vous devez déposer une demande sur le portail développeur de la STM en suivant la procédure suivante :

To use the STM API you will need a connection key. To obtain this, you must submit a request on the STM developer portal by following the following procedure :

- <https://www.stm.info/en/about/developers>

CFL - Canadian Football League

La ligue canadienne de football met à disposition plusieurs ensembles de données sur les matchs, les joueurs, les saisons, etc.

The Canadian Football League provides several sets of data on matches, players, seasons, etc.

Pour utiliser l'API de la CFL vous aurez besoin d'une clé de connexion. Pour obtenir celle-ci vous devez déposer une demande en cliquant sur "API key request form" sur la page suivante :

To use the CFL API you will need a login key. To obtain this, you must submit a request by clicking on "API key request form" on the following page :

- <https://api.cfl.ca/docs>

Autres API - Other API

Ce dépôt Github maintient une liste d'API publiques qui peuvent être utilisées avec ou sans authentification :

This Github repository maintains a list of public APIs that can be used with or without authentication :

- <https://github.com/public-apis/public-apis>

Données ouvertes - Open data

D'autres institutions mettent aussi à disposition des données que cela soit sous forme d'API ou via des fichiers à télécharger.

Other institutions also have open data, either in the form of APIs or via downloadable files.

- Ville de Montréal : <https://donnees.montreal.ca/>
- Québec : <https://www.donneesquebec.ca/>
- Canada : <https://open.canada.ca/en/open-data>

4. Git, GitHub

& GitLab

Git

Git est un système de contrôle de version distribué, gratuit et open source conçu pour gérer des petits aux très grands projets, avec rapidité et efficacité. Il détecte les changements que les utilisateurs ont effectué sur le code et emmagasine leurs codes sur une plateforme comme Github ou Gitlab.

Git is a free control system and openly distributed source designed to manage small to very large projects, with speed and efficiency. It detects the changes that users have made to the code and stores their codes on a platform like Github or Gitlab.

- <https://git-scm.com/video/what-is-git>
- <https://git-scm.com/video/get-going>

GitHub

Github permet aux développeurs de stocker et de partager le code qu'ils créent.

Github is a code hosting platform for version control and collaboration.

Chaque utilisateur peut contribuer aux projets mis en ligne sur Github en proposant des modifications.

It lets you and others work together on projects from anywhere.

- <https://docs.github.com/en/get-started/quickstart/hello-world>
- <https://www.freecodecamp.org/news/git-and-github-for-beginners/>

Atelier de PolyHx sur Git & GitHub -

PolyHx's workshop on Git & GitHub

https://www.youtube.com/watch?v=OL8r0aueAbI&ab_channel=PolyHx



GitLab

Gitlab est une plateforme de développement open source.

Elle couvre l'ensemble des étapes du Devops. Elle se base sur les fonctionnalités du logiciel Git, elle permet de piloter des dépôts de code source et de gérer leurs différentes versions.

GitLab is an open source end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more. Self-host GitLab on your own servers, in a container, or on a cloud provider.

- <https://about.gitlab.com/pricing/#gitlab-com>
- <https://github.com/SocialGouv/tutoriel-gitlab>
- <https://docs.gitlab.com>