Guide du développeur C/C++

ReadyDev

Plateforme de Développement Continu

Cours et Tutoriels adaptés au Sciences de l’Ingénieur

Produit par Gérard KESSE

# Sommaire

[Sommaire 2](#_Toc125294786)

[Mise en place d’un server TCP/IP (socket) 5](#_Toc125294787)

[Paramétrage du serveur 6](#_Toc125294788)

[Lancement du serveur 6](#_Toc125294789)

[Création du point de connexion 6](#_Toc125294790)

[Gestion de la connexion d’un client 7](#_Toc125294791)

[Affichage de la requête d’un client 7](#_Toc125294792)

[Lecture de la requête d’un client 8](#_Toc125294793)

[Fermeture du point de connexion 8](#_Toc125294794)

[Test sur le lancement du serveur (côté serveur) 8](#_Toc125294795)

[Test sur l’appel du serveur (côté client) 8](#_Toc125294796)

[Test sur l’affichage de la requête du client (côté serveur) 9](#_Toc125294797)

[Mise en place d’un server sécurisé TCP/IP (OpenSSL) 9](#_Toc125294798)

[Lancement du serveur 10](#_Toc125294799)

[Liens 10](#_Toc125294800)

[Mise en place d’un server HTTPS (Poco) 10](#_Toc125294801)

[Lancement du serveur 10](#_Toc125294802)

[Mise en place d’un server HTTPS (Poco) 10](#_Toc125294803)

[Création d’un serveur de SMS 11](#_Toc125294804)

[Apache 11](#_Toc125294805)

[Arrêt d’Apache sous Ubuntu 11](#_Toc125294806)

[Redémarrage d’Apache sous Ubuntu 12](#_Toc125294807)

[Configuration des logs d’un serveur Apache 12](#_Toc125294808)

[Affichage des logs du serveur web sous Ubuntu 12](#_Toc125294809)

[Liens 12](#_Toc125294810)

[C 13](#_Toc125294811)

[Liens 13](#_Toc125294812)

[C++ 13](#_Toc125294813)

[Liens 13](#_Toc125294814)

[Certbot 13](#_Toc125294815)

[Génération d’un certificat sous Ubuntu 13](#_Toc125294816)

[Suppression d’un certificat 13](#_Toc125294817)

[Affichage des certificats 14](#_Toc125294818)

[CURL 14](#_Toc125294819)

[Installation de CURL sous Windows 14](#_Toc125294820)

[Réalisation d’une requête HTTP GET 14](#_Toc125294821)

[DBeaver 16](#_Toc125294822)

[Connexion à une base de donnée SQLite 16](#_Toc125294823)

[Git 16](#_Toc125294824)

[Ajout de nouveaux fichiers 16](#_Toc125294825)

[Validation des modifications 16](#_Toc125294826)

[Pousser les modifications 17](#_Toc125294827)

[Récupération des modifications 17](#_Toc125294828)

[Sauvegarde des identifiants d’authentification 17](#_Toc125294829)

[Affichage de l’aide sur une commande git 17](#_Toc125294830)

[Afficher les configurations de base 17](#_Toc125294831)

[Ignorance d’un type de fichier 18](#_Toc125294832)

[Afficher toutes les branches à distance 18](#_Toc125294833)

[Afficher toutes les branches en local 18](#_Toc125294834)

[Changement de branche en local 18](#_Toc125294835)

[Non ignorance d’un fichier donné 19](#_Toc125294836)

[Annulation de toutes les modifications non validées 19](#_Toc125294837)

[Nettoyage des fichiers non validés 19](#_Toc125294838)

[Sauvegarde des fichiers modifiés ou rajoutés 20](#_Toc125294839)

[Résolution des problèmes d’authentification 20](#_Toc125294840)

[Résolution des problèmes de divergence 21](#_Toc125294841)

[Résolution des problèmes de CONFLICT/DELETE 21](#_Toc125294842)

[GitHub 21](#_Toc125294843)

[Génération d’un jeton 21](#_Toc125294844)

[Google Chrome 22](#_Toc125294845)

[Détection d’erreur de codage JavaScript 22](#_Toc125294846)

[Débogage d’un code JavaScript 23](#_Toc125294847)

[Débogage d’un code HTML 24](#_Toc125294848)

[JavaScript 24](#_Toc125294849)

[Encodage de données en base64 24](#_Toc125294850)

[Décodage de données en base64 25](#_Toc125294851)

[Résolution du problème d’erreur de syntaxe jeton non valide ou inattendu 25](#_Toc125294852)

[JsonC 25](#_Toc125294853)

[Installation de JsonC sous Ubuntu 25](#_Toc125294854)

[Développement avec JsonC 26](#_Toc125294855)

[Compilation avec JsonC sous Ubuntu 26](#_Toc125294856)

[Liens 26](#_Toc125294857)

[KeePass 26](#_Toc125294858)

[Installation de KeePass sous Windows 26](#_Toc125294859)

[LibXML2 26](#_Toc125294860)

[Récupération du contenu d’un nœud élément 26](#_Toc125294861)

[OpenSSL 27](#_Toc125294862)

[Installation d’OpenSSL sous Windows 27](#_Toc125294863)

[RaspberryPi 27](#_Toc125294864)

[Création d’un serveur de SMS 27](#_Toc125294865)

[Liens 27](#_Toc125294866)

[Sablotron 28](#_Toc125294867)

[Liens 28](#_Toc125294868)

[SSL for Free 28](#_Toc125294869)

[Obtention d’un certificat SSL 28](#_Toc125294870)

[ToirtoiseGit 29](#_Toc125294871)

[Changement de branche 29](#_Toc125294872)

[Suppression d’un fichier 30](#_Toc125294873)

[Sauvegarde des fichiers modifiés non validés 30](#_Toc125294874)

[Ubuntu 30](#_Toc125294875)

[Modification des permissions d’un fichier 30](#_Toc125294876)

[Modification du propriétaire et du groupe d’un fichier 31](#_Toc125294877)

[Création d’un nouveau screen 31](#_Toc125294878)

[Liste des screen en cours d’exécution 31](#_Toc125294879)

[Accès à un screen déjà créé 32](#_Toc125294880)

[Fermeture d’un screen déjà créé 32](#_Toc125294881)

[WampServer 32](#_Toc125294882)

[Installation de WampServer 32](#_Toc125294883)

[Démarrage de WampServer 33](#_Toc125294884)

[Redémarrage des services de WampServer 33](#_Toc125294885)

[Configuration d’une machine virtuelle dans WampServer 33](#_Toc125294886)

[Word 2007 35](#_Toc125294887)

[Activation du mode définir la langue 35](#_Toc125294888)

[XSLT 36](#_Toc125294889)

[Liens 36](#_Toc125294890)

[Fin 36](#_Toc125294891)

# Mise en place d’un server TCP/IP (socket)

Nous utiliserons la bibliothèque standard socket pour mettre en place un serveur TCP/IP.

## Paramétrage du serveur

void GSocketTest::runTest(int \_argc, char\*\* \_argv) {

GSocket lServer;

lServer.setModule("server\_tcp");

lServer.setHostname("0.0.0.0");

lServer.setPort(9081);

lServer.setBacklog(10);

lServer.setStartMessage("Démarrage du serveur...\n");

lServer.setStopMessage("Arrêt du serveur...\n");

lServer.run();

m\_logs.addLogs(lServer.getLogs());

}

## Lancement du serveur

bool GSocket::runServerTcp() {

if(!createSocketX()) return false;

printf("\n%s\n", m\_startMessage.c\_str());

struct sockaddr\_in lAddress2;

socklen\_t lSize = sizeof(lAddress2);

GThread lThread;

lThread.setThreadCB((void\*)*onThreadCB*);

while(1) {

m\_logs.showErrors();

m\_logs.clearMap();

GSocket\* lClient = new GSocket;

lClient->m\_socket = accept(m\_socket, (struct sockaddr\*)&lAddress2, &lSize);

if(lClient->m\_socket == -1) {

m\_logs.addError(GFORMAT("Erreur lors de l'acceptation de la connexion client.\n[%d]:%s", errno, strerror(errno)));

continue;

}

lThread.setParams((void\*)lClient);

if(!lThread.run()) {

m\_logs.addLogs(lThread.getLogs());

continue;

}

}

printf("\n%s\n", m\_stopMessage.c\_str());

closeSocket();

return !m\_logs.hasErrors();

}

## Création du point de connexion

bool GSocket::createSocket() {

if(m\_hostname.isEmpty()) {

m\_logs.addError("Erreur l'adresse n'est pas initialisée.");

return false;

}

if(!m\_port) {

m\_logs.addError("Erreur le port n'est pas initialisé.");

return false;

}

if(!m\_backlog) {

m\_logs.addError("Erreur le backlog n'est pas initialisé.");

return false;

}

m\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

if(m\_socket == -1) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du socket.\n[%d]:%s", errno, strerror(errno)));

return false;

}

struct sockaddr\_in lAddressIn;

bzero(&lAddressIn, sizeof(lAddressIn));

lAddressIn.sin\_family = AF\_INET;

lAddressIn.sin\_addr.s\_addr = inet\_addr(m\_hostname.c\_str());

lAddressIn.sin\_port = htons(m\_port);

int lBindOk = bind(m\_socket, (struct sockaddr\*)&lAddressIn, sizeof(lAddressIn));

if(lBindOk == -1) {

m\_logs.addError(GFORMAT("Erreur lors de la liaison du socket.\n[%d]:%s", errno, strerror(errno)));

return false;

}

int lListenOk = listen(m\_socket, m\_backlog);

if(lListenOk == -1) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du backlog.\n[%d]:%s", errno, strerror(errno)));

return false;

}

return !m\_logs.hasErrors();

}

## Gestion de la connexion d’un client

void\* GSocket::onThreadCB(void\* \_params) {

GSocket\* lClient = (GSocket\*)\_params;

lClient->runThreadCB();

lClient->closeSocket();

delete lClient;

return 0;

}

## Traitement de la requête d’un client

bool GSocket::runThreadCB() {

GString lRequest;

readData(lRequest);

GLOGT(eGMSG, "RECEPTION [%d] :\n%s\n", lRequest.size(), lRequest.c\_str());

sendEchoHttp();

return !m\_logs.hasErrors();

}

## Lecture de la requête d’un client

bool GSocket::readData(GString& \_dataOut, int \_size) {

if(\_size <= 0) return false;

char lBuffer[*BUFFER\_SIZE* + 1];

int lSize = 0;

while(1) {

int lBytes = recv(m\_socket, lBuffer, *BUFFER\_SIZE*, 0);

if(lBytes <= 0) return false;

lBuffer[lBytes] = 0;

\_dataOut += lBuffer;

lSize += lBytes;

if(lSize >= \_size) return true;

}

return true;

}

## Traitement de la réponse d’un client

bool GSocket::sendEchoHttp() {

GHttp lHttp;

lHttp.setModule("response");

lHttp.setContentText("Bonjour tout le monde !");

lHttp.run();

m\_logs.addLogs(lHttp.getLogs());

GString lResponse = lHttp.getResponseText();

GLOGT(eGMSG, "EMISSION [%d] :\n%s\n", lResponse.size(), lResponse.c\_str());

sendData(lResponse);

return !m\_logs.hasErrors();

}

## Envoi de la réponse d’un client

bool GSocket::sendData(const GString& \_dataIn) {

if(\_dataIn.isEmpty()) return true;

int lSize = \_dataIn.size();

const char\* lBuffer = \_dataIn.c\_str();

int lIndex = 0;

while(1) {

int lBytes = send(m\_socket, &lBuffer[lIndex], lSize - lIndex, 0);

if(lBytes <= 0) break;

lIndex += lBytes;

if(lIndex >= lSize) break;

}

return !m\_logs.hasErrors();

}

## Fermeture du point de connexion

void GSocket::closeSocket() {

m\_logs.showErrors();

m\_logs.clearMap();

if(m\_socket > 0) {close(m\_socket); m\_socket = 0;}

}

## Test sur le lancement du serveur (côté serveur)

Cela suppose que vous êtes sur la machine serveur.

Ouvrir un terminal. Lancer le serveur.



## Test sur l’appel du serveur (côté client)

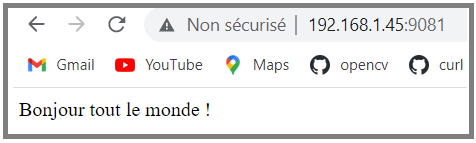
Cela suppose que vous avez déjà lancé le serveur. Vous êtes sur la machine client. Vous disposez d’un navigateur web pour appeler le serveur.

Ouvrir le navigateur web. Saisir l’adresse du serveur. Appeler le serveur.

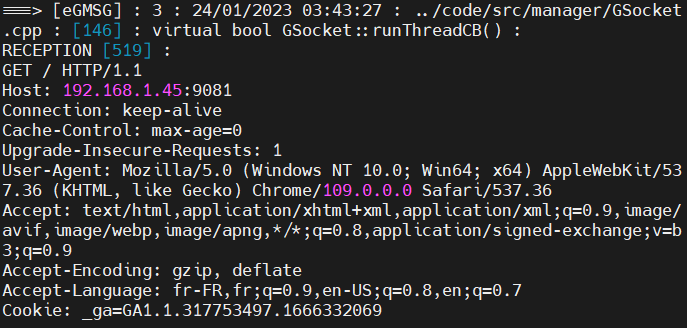
**GOOGLE\_CHROME**

**Barre d’adresse >** [**http://192.168.1.45:9081/**](http://192.168.1.45:9081/)

**Entrer**

****

## Test sur la réception de la requête du client (côté serveur)

****

## Test sur l’envoi de la réponse du client (côté serveur)

****

## Liens

Obtenir le nombre d'octets qui sont immédiatement disponibles pour la lecture (ioctl).

<https://github.com/kristrev/tproxy-example/blob/master/tproxy_example.c>

# Mise en place d’un server sécurisé TCP/IP (OpenSSL)

Nous utiliserons la bibliothèque OpenSSL pour mettre en place un serveur sécurisé TCP/IP.

## Initialisation du serveur

**int** main(**int** \_argc, **char**\*\* \_argv) {

GOpenSSL().initModule();

GProcess lProcess;

lProcess.initProcess();

lProcess.run(\_argc, \_argv);

lProcess.showErrors();

GOpenSSL().clearModule();

**return** 0;

}

## Initialisation du module OpenSSL

void GOpenSSL::initModule() {

SSL\_load\_error\_strings();

ERR\_load\_crypto\_strings();

OpenSSL\_add\_all\_algorithms();

SSL\_library\_init();

}

## Nettoyage du module OpenSSL

void GOpenSSL::clearModule() {

ERR\_free\_strings();

EVP\_cleanup();

}

## Paramétrage du serveur

void GOpenSSLTest::runTest(int \_argc, char\*\* \_argv) {

GOpenSSL lOpenSSL;

lOpenSSL.setStartMessage("Démarrage du serveur...\n");

lOpenSSL.setStopMessage("Arrêt du serveur...\n");

lOpenSSL.setHostname("0.0.0.0");

lOpenSSL.setPort(9081);

lOpenSSL.setBacklog(10);

lOpenSSL.setCountry("FR");

lOpenSSL.setCommonName("ReadyDev");

lOpenSSL.setPrivateKeyFile("certificates/privkey.pem");

lOpenSSL.setCertificateFile("certificates/fullchain.pem");

lOpenSSL.setHasGenerateCertificate(false);

lOpenSSL.run();

m\_logs.addLogs(lOpenSSL.getLogs());

}

## Lancement du serveur

bool GOpenSSL::run() {

GSocket lSocket;

lSocket.setHostname(m\_hostname);

lSocket.setPort(m\_port);

lSocket.setBacklog(m\_backlog);

if(!lSocket.createSocket()) {

m\_logs.addLogs(lSocket.getLogs());

return false;

}

m\_socket = lSocket.getSocket();

struct sockaddr\_in lClientAddress;

socklen\_t lClientAddressSize = sizeof(lClientAddress);

if(!initSSL()) return false;

GThread lThread;

lThread.setThreadCB((void\*)*onThreadCB*);

printf("\n%s\n", m\_startMessage.c\_str());

while(1) {

m\_logs.showErrors();

m\_logs.clearMap();

GOpenSSL\* lClient = new GOpenSSL;

lClient->m\_socket = accept(m\_socket, (struct sockaddr\*)&lClientAddress, &lClientAddressSize);

if(lClient->m\_socket <= 0) {

m\_logs.addError(GFORMAT("Erreur lors de l'acceptation de la connexion client.\n[%lu]:%s", errno, strerror(errno)));

continue;

}

lClient->setOpenSSL(\*this);

lThread.setParams((void\*)lClient);

if(!lThread.run()) {

m\_logs.addLogs(lThread.getLogs());

continue;

}

}

printf("\n%s\n", m\_stopMessage.c\_str());

cleanSSL();

return !m\_logs.hasErrors();

}

## Création du point de connexion standard

bool GSocket::createSocket() {

if(m\_hostname.isEmpty()) {

m\_logs.addError("Erreur l'adresse n'est pas initialisée.");

return false;

}

if(!m\_port) {

m\_logs.addError("Erreur le port n'est pas initialisé.");

return false;

}

if(!m\_backlog) {

m\_logs.addError("Erreur le backlog n'est pas initialisé.");

return false;

}

m\_socket = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

if(m\_socket == -1) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du socket.\n[%d]:%s", errno, strerror(errno)));

return false;

}

struct sockaddr\_in lAddressIn;

bzero(&lAddressIn, sizeof(lAddressIn));

lAddressIn.sin\_family = AF\_INET;

lAddressIn.sin\_addr.s\_addr = inet\_addr(m\_hostname.c\_str());

lAddressIn.sin\_port = htons(m\_port);

int lBindOk = bind(m\_socket, (struct sockaddr\*)&lAddressIn, sizeof(lAddressIn));

if(lBindOk == -1) {

m\_logs.addError(GFORMAT("Erreur lors de la liaison du socket.\n[%d]:%s", errno, strerror(errno)));

return false;

}

int lListenOk = listen(m\_socket, m\_backlog);

if(lListenOk == -1) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du backlog.\n[%d]:%s", errno, strerror(errno)));

return false;

}

return !m\_logs.hasErrors();

}

## Initialisation du certificat et de la clé privée

bool GOpenSSL::initSSL() {

m\_context = SSL\_CTX\_new(SSLv23\_server\_method());

if(!m\_context) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du contexte.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

SSL\_CTX\_set\_options(m\_context, SSL\_OP\_SINGLE\_DH\_USE);

if(m\_hasGenerateCertificate) {

if(!generatePrivateKey()) return false;

if(!generateCertificate()) return false;

int lCertificateOk = SSL\_CTX\_use\_certificate(m\_context, m\_certificate);

if(lCertificateOk <= 0) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du certificat.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

SSL\_CTX\_set\_default\_passwd\_cb(m\_context, *onPasswordCB*);

int lPrivateKeyOk = SSL\_CTX\_use\_PrivateKey(m\_context, m\_privateKey);

if(lPrivateKeyOk <= 0) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation de la clé privée.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

}

else {

int lCertificateOk = SSL\_CTX\_use\_certificate\_file(m\_context, m\_certificateFile.c\_str() , SSL\_FILETYPE\_PEM);

if(lCertificateOk <= 0) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du certificat.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

SSL\_CTX\_set\_default\_passwd\_cb(m\_context, *onPasswordCB*);

int lPrivateKeyOk = SSL\_CTX\_use\_PrivateKey\_file(m\_context, m\_privateKeyFile.c\_str(), SSL\_FILETYPE\_PEM);

if(lPrivateKeyOk <= 0) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation de la clé privée.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

}

SSL\_CTX\_set\_verify(m\_context, SSL\_VERIFY\_NONE, 0);

return !m\_logs.hasErrors();

}

## Gestion de la connexion d’un client

void\* GOpenSSL::onThreadCB(void\* \_params) {

GOpenSSL\* lClient = (GOpenSSL\*)\_params;

lClient->acceptSSL();

lClient->runThreadCB();

lClient->closeSSL();

delete lClient;

return 0;

}

## Création du point de connexion sécurisé

bool GOpenSSL::acceptSSL() {

m\_ssl = SSL\_new(m\_context);

if(!m\_ssl) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du socket sécurisé.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

int lSetOk = SSL\_set\_fd(m\_ssl, m\_socket);

if(lSetOk <= 0) {

m\_logs.addError(GFORMAT("Erreur lors de l'initialisation du descripteur.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

int lAcceptOk = SSL\_accept(m\_ssl);

if(lAcceptOk <= 0) {

m\_logs.addError(GFORMAT("Erreur lors de l'acceptation de la connexion sécurisée.\n[%lu]:%s", ERR\_get\_error(), ERR\_error\_string(ERR\_get\_error(), NULL)));

return false;

}

return !m\_logs.hasErrors();

}

## Traitement de la requête d’un client

bool GOpenSSL::runThreadCB() {

if(m\_logs.hasErrors()) return false;

GString lRequest;

readData(lRequest);

GLOGT(eGMSG, "RECEPTION [%d] :\n%s\n", lRequest.size(), lRequest.c\_str());

sendEchoHttp();

return true;

}

## Lecture de la requête d’un client

bool GOpenSSL::readData(GString& \_dataOut) {

char lBuffer[*BUFFER\_SIZE* + 1];

int lSize = 0;

while(1) {

int lBytes = SSL\_read(m\_ssl, lBuffer, *BUFFER\_SIZE*);

if(lBytes <= 0) break;

if(lSize >= *BUFFER\_MAX*) {

m\_logs.addError("Erreur le nombre d'octets maximal a été atteint.");

break;

}

lBuffer[lBytes] = 0;

\_dataOut += lBuffer;

int lPending = SSL\_pending(m\_ssl);

if(lPending <= 0) break;

}

return !m\_logs.hasErrors();

}

## Traitement de la réponse d’un client

bool GOpenSSL::sendEchoHttp() {

GHttp lHttp;

lHttp.setModule("response");

lHttp.setContentText("Bonjour tout le monde !");

lHttp.run();

m\_logs.addLogs(lHttp.getLogs());

GString lResponse = lHttp.getResponseText();

GLOGT(eGMSG, "EMISSION [%d] :\n%s\n", lResponse.size(), lResponse.c\_str());

sendData(lHttp.getResponseText());

return !m\_logs.hasErrors();

}

## Envoi de la réponse d’un client

bool GOpenSSL::sendData(const GString& \_dataIn) {

if(\_dataIn.isEmpty()) return false;

int lSize = \_dataIn.size();

const char\* lBuffer = \_dataIn.c\_str();

int lIndex = 0;

while(1) {

int lBytes = SSL\_write(m\_ssl, &lBuffer[lIndex], lSize - lIndex);

if(lBytes <= 0) return false;

lIndex += lBytes;

if(lIndex >= lSize) break;

}

return true;

}

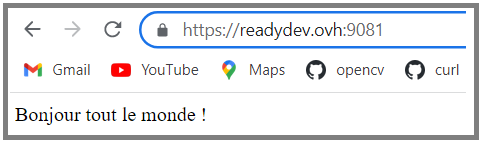
## Test sur le lancement du serveur (côté serveur)

Cela suppose que vous êtes sur la machine serveur.

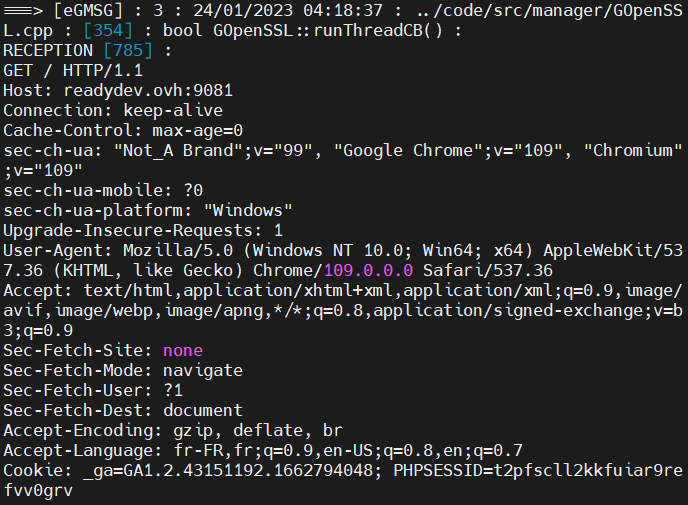
Ouvrir un terminal. Lancer le serveur.



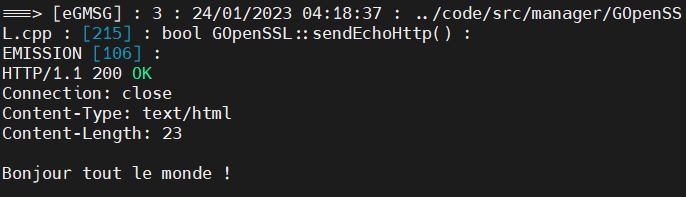
## Test sur l’appel du serveur (côté client)



## Test sur la réception de la requête d’un client (côté serveur)



## Test sur l’envoi de la réponse d’un client (côté serveur)



## Test sur la réception d’une requête non sécurisée (côté serveur)

## Liens

Exemple de serveur OpenSSL

<https://gist.github.com/amir-saniyan/5c90f0098b171c5fc01fa143c1dc43ad>

Serveur OpenSSL, exemple de référence

<https://gist.github.com/MartinMReed/6393150>

Génère un certificat x509 auto-signé à l'aide d'OpenSSL.

<https://gist.github.com/nathan-osman/5041136?permalink_comment_id=1751573>

Exemple OpenSSL simple qui fait une requête HTTPS

<https://gist.github.com/rexim/ab45940928399bbda212185b4a6a6339>

Exemple compact d'utilisation d'OpenSSL avec des clés/certificats auto-signés (sans mot de passe), DTLS et BIOS de mémoire

<https://gist.github.com/roxlu/9835067>

Qu'est-ce que HTTP Keep-Alive

<https://www.imperva.com/learn/performance/http-keep-alive/>

<https://www.stackpath.com/edge-academy/what-is-keep-alive/>

La première ligne d'un message de réponse est la ligne d'état, composée de la version du protocole suivie d'un code d'état numérique et de sa phrase textuelle associée, chaque élément étant séparé par des caractères SP. Aucun CR ou LF n'est autorisé sauf dans la séquence CRLF finale.

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html>

Obtenir le nombre d'octets lisibles mis en mémoire tampon dans un objet SSL (SSL\_pending).

<https://github.com/Andersbakken/openssl-examples/blob/master/read_write.c>

Obtenir ou définir l'adresse de diffusion d'un appareil. Obtenir ou définir le masque de réseau d'un appareil (ioctl).

<https://gist.github.com/goog/f4e84613f011c482bb8c>

# Mise en place d’un server HTTPS (Poco)

2ous utiliserons Poco pour gérer les demandes des clients côté serveur via une communication sécurisée HTTPS.

Cela suppose que vous disposiez d’un client capable d’émettre des demandes GET et POST via une communication sécurisée HTTPS.

## Lancement du serveur

# Mise en place d’un server HTTPS (Poco)

2ous utiliserons Poco pour gérer les demandes des clients côté serveur via une communication sécurisée HTTPS.

Cela suppose que vous disposiez d’un client capable d’émettre des demandes GET et POST via une communication sécurisée HTTPS.

## Création d’un serveur de SMS

Pour pouvoir envoyer des SMS à partir de votre application. Cela suppose que vous disposez d’une clé WIFI avec intégration de carte SIM. Vous êtes déjà abonné chez un opérateur mobile.

# Mise en place d’un contrôle d’entrées/sortie (ioctl)

## Liens

Exemple simple sur la façon de créer un pilote IOCTL pour Linux. Cela peut être utilisé à des fins de test simples.

<https://github.com/pokitoz/ioctl_driver>

# Apache

## Arrêt d’Apache sous Ubuntu

Cela suppose que le fonctionnement d’Apache empêche l’exécution d’une opération.

Arrêter Apache.

**sudo systemctl stop apache2**

## Redémarrage d’Apache sous Ubuntu

Cela suppose que vous avez une opération qui nécessite le fonctionnement d’Apache.

Redémarrer Apache.

**sudo systemctl restart apache2**

## Configuration des logs d’un serveur Apache

Cela suppose que vous avez constaté des disfonctionnements dans votre serveur web.

Configurer les logs d’un serveur web. Créer le répertoire des logs. Configurer les logs le fichier de configuration des machines virtuelles.

**WEBROOT--> /home/gkesse/Programs/ReadyDev**

**LOGS-----> /home/gkesse/Programs/ReadyDev/logs**

**sudo nano /etc/apache2/sites-available/000-default.conf**

**ErrorLog WEBROOT/logs/error.log**

**CustomLog WEBROOT/logs/access.log combined**

## Affichage des logs du serveur web sous Ubuntu

Cela suppose que vous avez configuré les logs dans votre serveur web. Vous constatez un disfonctionnement de votre serveur web et que vous voulez déboguer votre serveur web.

Afficher les logs du serveur web.

**tail -f /home/gkesse/Programs/ReadyDev/logs/error.log**

## Liens

Fonctionalités de Base et Modules Multi-Processus (MPM)

<https://httpd.apache.org/docs/2.4/fr/mod/>

# C

## Liens

Une liste organisée de frameworks C impressionnants, de bibliothèques, de ressources et d'autres choses brillantes. Inspiré par tous les autres projets géniaux... qui existent.

<https://github.com/oz123/awesome-c>

# C++

## Liens

Une liste organisée de frameworks C++ (ou C) impressionnants, de bibliothèques, de ressources et de choses brillantes. Inspiré par des trucs géniaux.

<https://github.com/fffaraz/awesome-cpp>

# Certbot

## Génération d’un certificat sous Ubuntu

Cela suppose que vous disposez d’un nom de domaine valide et que vous cherchez à sécuriser les communications avec votre serveur web.

Ouvrir un terminal. Saisir la commande de génération de certificat.

**sudo certbot certonly \**

**--standalone \**

**--agree-tos \**

**--no-eff-email \**

**-d readydev.ovh \**

**-d www.readydev.ovh \**

## Suppression d’un certificat

Cela suppose que vous générer plusieurs certificats pour un même nom de domaine et que vous cherchez à faire un coup de propre.

Supprimer un certificat.

**sudo certbot delete**

**Numéro du certificat > 1**

**Confirmation > Y**

## Affichage des certificats

Cela suppose que vous voulez avoir un aperçu sur tous les certificats générés sur votre machine.

Afficher les certificats.

**sudo certbot certificates**

# CURL

## Installation de CURL sous Windows

Cela suppose que vous voulez vous lancer dans la communication réseau sous Windows.

Installer CURL sous Windows. Télécharger CURL pour Windows. Décompresser CURL. Copier/Coller les fichiers de développement de CURL vers le répertoire de développement.

**WEB >** [**https://curl.se/windows/**](https://curl.se/windows/)

**Télécharger > curl for 64-bit**

**CURL\_DIR > bin > DEV\_DIR**

**CURL\_DIR > lib > DEV\_DIR**

**CURL\_DIR > include > DEV\_DIR**

## Réalisation d’une requête HTTP GET

Cela suppose que vous avez configuré les fichiers de développement de CURL dans votre répertoire de développement.

Réaliser une requête HTTP GET.

Initialiser la configuration de base.

**curl\_global\_init(CURL\_GLOBAL\_ALL);**

**CURL\* lCurl = curl\_easy\_init();**

**curl\_easy\_setopt(\_curl, *CURLOPT\_HTTPGET*, 1L);**

**curl\_easy\_setopt(\_curl, *CURLOPT\_HTTP\_VERSION*, *CURL\_HTTP\_VERSION\_1\_1*);**

**curl\_easy\_setopt(\_curl, *CURLOPT\_ERRORBUFFER*, lError);**

**curl\_easy\_setopt(\_curl, *CURLOPT\_URL*, m\_fullUrl.c\_str());**

**curl\_easy\_setopt(\_curl, *CURLOPT\_FOLLOWLOCATION*, 1L);**

**curl\_easy\_setopt(\_curl, *CURLOPT\_WRITEFUNCTION*, *onWrite*);**

**curl\_easy\_setopt(\_curl, *CURLOPT\_WRITEDATA*, &lBuffer);**

**curl\_easy\_setopt (\_curl, *CURLOPT\_VERBOSE*, 0L);**

Définir un nom d’utilisateur et un mot de passe si nécessaire.

**curl\_easy\_setopt(\_curl, *CURLOPT\_USERNAME*, m\_apiUsername.c\_str());**

**curl\_easy\_setopt(\_curl, *CURLOPT\_PASSWORD*, m\_apiPassword.c\_str());**

Ajouter une entête si nécessaire.

**GString lKey = "Content-Type";**

**GString lValue = "application/xml";**

**GString lData = GFORMAT("%s: %s", lKey.c\_str(), lValue.c\_str());**

**\_headers = curl\_slist\_append(\_headers, lData.c\_str());**

**curl\_easy\_setopt(\_curl, *CURLOPT\_HTTPHEADER*, lHeaders);**

Exécuter la requête.

**CURLcode lCurlOk = curl\_easy\_perform(\_curl);**

Récupérer le code de retour.

**curl\_easy\_getinfo(\_curl, *CURLINFO\_RESPONSE\_CODE*, &m\_responseCode);**

Libérer les ressources.

**curl\_easy\_cleanup(\_curl);**

**curl\_slist\_free\_all(lHeaders);**

**curl\_global\_cleanup();**

Récupérer la réponse.

**\_response = lBuffer;**

Définir la fonction de rappel d’écriture dans le buffer.

**if(\_writerData == NULL) return 0;**

**\_writerData->append(\_data, \_size \* \_nmemb);**

**return \_size \* \_nmemb;**

# DBeaver

## Connexion à une base de donnée SQLite

Cela suppose que vous voulez désormais passer par un outil graphique pour gérer vos bases de données SQLite.

Ouvrir la fiche SQLite. Définir le chemin de la base de données. Créer la connexion à la base de données SQLite.

**Fichiers > Nouveau**

**DBeaver > Connexion base de données**

**Suivant**

**SQLite**

**Suivant**

**Général > Connect by > Host**

**Path > C:\chemin\database.sql**

**Terminer**

# Git

## Ajout de nouveaux fichiers

Cela suppose que vous avez ajouté des nouveaux fichiers dans votre dépôt local et que vous souhaitez suivre un ou plusieurs fichiers voire tous les nouveaux fichiers. C’est la phase de suivi de fichier.

Suivre un fichier.

**git add mon\_fichier.txt**

**git add mon\_fichier\_01.txt mon\_fichier\_02.txt**

**git add --all**

## Validation des modifications

Cela suppose que vous êtes satisfait de vos modifications et que vous voulez les valider. C’est la phase de validation ou d’enregistrement des modifications.

Valider les modifications.

**git commit –m "Initial Commit"**

## Pousser les modifications

Cela suppose que vous avez enregistré les modifications.

Pousser les modifications.

**git push**

## Récupération des modifications

Cela suppose que vous voulez récupérer les modifications sur le référentiel.

Récupérer les modifications.

**git pull**

## Sauvegarde des identifiants d’authentification

Cela suppose que vous en avez assez de saisir les identifiants d’authentification git à chaque que vous réalisez un push vers le référentiel git.

**git config --global credential.helper store**

## Affichage de l’aide sur une commande git

Cela suppose que vous voulez plus d’informations sur l’utilisation d’une commande git.

Afficher l’aide sur une commande git.

**git help rebase**

## Afficher les configurations de base

Cela suppose que vous voulez des informations sur l’utilisateur d’un dépôt local.

Afficher les configurations de base.

**git config --list**

## Ignorance d’un type de fichier

Cela suppose que vous ne vouliez surtout pas pousser vers le référentiel des fichiers que vous considérez comme des fichiers intermédiaires tels que les fichiers objet (.obj, .o) générés lors de la compilation d’un projet C/C++.

Ignorer un type de fichier. Aller dans le dépôt local git. Ouvrir le fichier de configuration gitignore. Ignorer le type de fichier.

**GIT > .gitignore**

**/\*\*/\*.obj**

**/\*\*/\*.o**

**/\*\*/\*.docx**

## Afficher toutes les branches à distance

Cela suppose que vous voulez changer de branche et que vous aimerez avoir un aperçu sur toutes les branches à distance.

Afficher toutes les branches à distance.

**git branch -r**

## Afficher toutes les branches en local

Cela suppose que vous voulez changer de branche et que vous aimerez avoir un aperçu sur toutes les branches en local.

Afficher toutes les branches en local.

**git branch -a**

## Changement de branche en local

Cela suppose que vous êtes intéressés par des modifications sur une autre branche.

Changer de branche.

**git checkout 1.0**

Changer de branche et créer la branche si elle n’existe.

**git checkout –b 1.0**

## Non ignorance d’un fichier donné

Cela suppose que vous avez ignoré un ensemble de fichiers donné, et que néanmoins, vous souhaitez quand même conserver un fichier spécifique parmi cet ensemble.

Ne pas ignorer un fichier donné. Aller dans le dépôt local git Ouvrir le fichier de configuration gitignore. Informer git de ne pas ignorer le fichier donné.

**GIT > .gitignore**

**/\*\*/\*.obj**

**/\*\*/\*.o**

**/\*\*/\*.docx**

**!/guide/guide.docx**

## Annulation de toutes les modifications non validées

Cela suppose que vous travaillez sur un serveur de tests qui vous permet de vérifier le bon fonctionnement de vos modifications sur une branche donnée sans les valider. Avant de basculer sur une nouvelle le serveur doit d’abord annuler les anciennes modifications non validées afin d’éviter des conflits.

Annuler les modifications. Vérifier le statut du dépôt.

**git checkout .**

**git status**

## Nettoyage des fichiers non validés

Cela suppose que vous avez transféré vos modifications sur un serveur de tests afin de vérifier le bon fonctionnement de vos modifications sur une branche donnée. Vous voulez désormais changer de branche de travail ou récupérer de nouvelles modifications sur la branche en cours sans rencontrer de conflits. Vous devez donc nettoyer votre répertoire de travail avant de procéder à ces opérations.

Nettoyer le répertoire de travail. Récupérer de nouvelles modifications. Changer de branche.

**git clean -df**

## Sauvegarde des fichiers modifiés ou rajoutés

Cela suppose que vous voulez réduire le temps de transfert de vos modifications vers une machine distante en vue de réaliser des tests. Plus tôt que de transférer tout votre dépôt local en raison de quelques modifications, ce qui n’est pas nécessaire, vous voulez désormais ne transférer que vos modifications.

Aller dans votre dépôt local. Récupérer le résultat de la commande simplifiée sur les statuts. Parcourir le résultat fichier par fichier. Copier chaque fichier avec son chemin complet dans un répertoire de votre choix.

**@echo off**

**::===============================================**

**set "GPWD=%cd%"**

**set "GREPO=C:\Users\tiaka\Downloads\eTools\ePrograms\ReadyCode"**

**::===============================================**

**cd %GREPO%**

**::===============================================**

**setlocal enabledelayedexpansion**

**for /f "delims=" %%i in ('git status -s') do (**

**set "tmp=%%i"**

**set "tmp=!tmp:~2!"**

**set "tmp=!tmp:/=\!"**

**set "tmp=!tmp:~1!**

**echo f | xcopy !tmp! z\_dev\!tmp!**

**)**

**endlocal**

**::===============================================**

**cd %GPWD%**

**::===============================================**

## Résolution des problèmes d’authentification

Cela suppose que vous avez essayé de pousser vos modifications vers le référentiel github et que github vous a demandé vos identifiants d’authentification. Vous avez saisi votre nom d’utilisateur et votre mot de passe github, mais hélas, rien d’intéressant, l’opération a échoué.

Sachez que github a modifier sa méthode d’authentifiaction. La méthode username password fait partie de l’histoire ancienne. GitHub utilise désormais la méthode token pour l’authentification.

Résoudre le problème. Générer un jéton github. Pousser vos modifiactions. Saisir votre nom d’utilisateur. Saisir votre jeton github.

**git fetch**

**git push**

**username > nom\_d\_utilisateur**

**password > mon\_token\_github**

## Résolution des problèmes de divergence

Cela suppose qu’un fichier FA a été modifié sur la machine MA et poussé sur le référentiel RA alors qu’il a été modifié sur la machine MB avant d’être récupéré depuis le référentiel RA. Vous tentez de récupérer les évolutions du référentiels RA sur la machine MB et vous observez une divergence de branches. Cependant, vous ne voulez surtout pas que vos modifications sur la machine MB fassent l’objet d’une nouvelle branche.

Résoudre le problème. Récupérer les évolutions. Réappliquer vos modifications par-dessus les modifications.

**git pull –-rebase**

## Résolution des problèmes de CONFLICT/DELETE

Cela suppose qu’un fichier FA a été supprimé sur la machine MA et poussé sur le référentiel RA alors qu’il a été modifié sur la machine MB avant d’être récupéré depuis le référentiel RA.

Résoudre le problème. Supprimer le fichier FA sur la machine MB. Réappliquer les commits CB sur les commits CA. Pousser les modifications.

**MB > git rm FA**

**MB > git rebase –continue**

**MB > git push**

# GitHub

## Génération d’un jeton

Cela suppose que vous avez un dépôt git qui tente de pousseur vos modifications sur votre référentiel github. Vous devez fournir des identifiants d’authentification pour réaliser l’opération.

Générer un jeton.

**GITHUB > Paramètres > Paramètres du développeur**

**Jeton d’accès personnels > Jetons (classiques)**

**Générer un jeton**

**Expiration > Pas d’espiration**

**Générer le jeton**

**Copier le jeton**

Régénérer un jeton.

**GITHUB > Paramètres > Paramètres du développeur**

**Jeton d’accès personnels > Jetons (classiques)**

**Sélectionner un jeton > git\_token**

**Régénérer le jeton**

**Expiration > Pas d’espiration**

**Régénérer le jeton**

**Copier le jeton**

# Google Chrome

## Détection d’erreur de codage JavaScript

Cela suppose que vous en avez assez de déboguer manuel votre code JavaScript via un éditeur de texte ou par la méthode de parcours de scripts. Vous voulez désormais détecter automatiquement les erreurs de codage de votre script au moment de son exécution.

Ouvrir la page web à déboguer.

[**http://localhost:8001/home/admin**](http://localhost:8001/home/admin)

Ouvrir l’outil de développement.

**GOOGLE\_CHROME > Personnaliser et contrôler Google Chrome**

**Plus d’outils > Outils de développement**

Ouvrir la liste des erreurs si présentes.

**Open Console to view 2 erroros**

Identifier les erreurs de fichiers non trouvés si présentes.

**> GET** [**http://localhost:8001/js/class/GObjectUi.js**](http://localhost:8001/js/class/GObjectUi.js)

**net::ERR\_ABORTED 404 (Not Found)**

Identifier les erreurs de références incorrectes.

**> Uncaught ReferenceError: GEditorUi is not defined**

**at GEditorUi.js:148:17**

Cliquer sur le fichier pour afficher le code érroné.

**Erreur-----> var lEditorUi = new GEditorHTML();**

**Correctif--> var lEditorUi = new GEditorUi();**

## Débogage d’un code JavaScript

Cela suppose que vous en avez assez de déboguer manuel votre code JavaScript via un éditeur de texte ou par la méthode d’affichage en console. Vous voulez désormais connaître l’état et le contenu des différentes variables de votre script au moment de l’exécution.

Ouvrir la page à déboguer. Ouvrir l’outil de développement. Sélectionner le script à déboguer. Placer un point d’arrêt. Actualiser la page à déboguer. Déboguer le script.

**Ouvrir >** [**http://localhost:8001/home/admin**](http://localhost:8001/home/admin)

**Personnaliser et contrôler Google Chrome**

**Plus d’outils > Outils de développement**

**Sources > Page**

**js > class > GServer.js**

**Cliquer sur un numéro > 107 > pour placer un point d’arrêt**

**Rafraichir >** [**http://localhost:8001/home/admin**](http://localhost:8001/home/admin)

**F8 > Reprendre l’exécution du script > pour passer au point d’arrêt suivant**

**F10 > Passer l’appel de fonction suivant > pour avancer d’un pas**

**F11 > Passer à l’appel de fonction suivant > pour entrer dans la fonction actuelle**

**Shift + F11 > Quitter la fonction actuelle > pour quitter la fonction actuelle**

Afficher le contenu d’une variable. Placer le curseur au-dessus de la variable. Observer le contenu de la variable.

**ma\_variable > placer le curseur au-dessus > observer le contenu**

## Débogage d’un code HTML

Cela suppose que vous avez un mauvais rendu de votre page web à cause des erreurs de codage tels que des blocs de div imbriqués par erreur et vous avez du mal à trouver la section de code en erreur. Peu importe le langage que vous utilisez côté serveur, le rendu final de votre page web sera toujours en HTML. En identifiant, le bloc de code HTML erroné vous pouvez retourner vers votre serveur pour effectuer les correctifs nécessaires.

Ouvrir la page à déboguer.

[**http://localhost:8001/home/admin**](http://localhost:8001/home/admin)

Ouvrir l’outil de développement.

**Personnaliser et contrôler Google Chrome**

**Plus d’outils > Outils de développement**

Ouvrir la liste des élements. Cliquer sur les blocs de div. Etablir les correspondances entre les blocs de div et le rendu HTML. Identifier le bloc de code erroné. Dans l’exemple qui suit on est censé aligner 4 blocs de div MainBlock. On constate lors du rendu que le 4ème bloc à l’air décalé par rapport aux 3 premiers blocs. Problème : Le 4ème est imbriqué dans le 2ème bloc.

**Eléments >**

**>---<div class="MainBlock">---</div>**

**>---<div class="MainBlock">---</div>**

**>---<div class="MainBlock">**

**|-------<div class="Content">---</div>**

**|-------<div class="MainBlock">---</div>**

**|---</div>**

# JavaScript

## Encodage de données en base64

Cela suppose que voulez transmettre des données sur le réseau sans vous soucier des problèmes de conversion. Vous avez besoin d’encoder les données en base64 avant de les transmettre.

Appeler la méthode d’encodage en base64.

**var lBase64 = btoa(lData);**

## Décodage de données en base64

Cela suppose que voulez transmettre des données sur le réseau sans vous soucier des problèmes de conversion. Vous avez besoin d’encoder les données en base64 avant de les transmettre.

Appeler la méthode d’encodage en base64.

**var lData = atob(lBase64);**

## Résolution du problème d’erreur de syntaxe jeton non valide ou inattendu

Cela suppose que vous avez appelé une fonction JavaScript en lui fournissant un argument défini sur plusieurs lignes. Cela se produit en général lorsqu’on crée le message côté serveur pour l’exécuter en JavaScript.

Insérer plus tôt la donnée dans une balise input hidden. Associer un Id à la balise. Accéder à la donnée en JavaScript via l’Id.

Côté serveur en PHP.

**echo *sprintf*("<input type='hidden' id='LogsPhpData' value='%s'/>\n", $this->serialize());**

**echo *sprintf*("<script>call\_server('logs', 'show\_php\_logs');</script>\n");**

Côté client en JavaScript.

**var lLogsPhpData = document.getElementById("LogsPhpData");**

**var lData = lLogsPhpData.value;**

**this.loadLogs(lData);**

# JsonC

## Installation de JsonC sous Ubuntu

Cela suppose que vous voulez manipuler des données au format JSON.

Ouvrir un terminal. Installer JsonC.

**sudo apt install libjson-c-dev**

**sudo apt install libjson-c5**

## Développement avec JsonC

Cela suppose que vous voulez réaliser un programme C/C++ avec JSON.

Ajouter le fichier header Json-C.

**#include <json-c/json.h>**

## Compilation avec JsonC sous Ubuntu

Cela suppose que vous voulez déjà développé votre code avec Json-C en C/C++. Vous voulez procéder à la compilation avec gcc/g++.

Ajouter le linker Json-C.

**-ljson-c**

## Liens

Comment installer Json-C sous Linux (Ubuntu) ?

<https://linuxprograms.wordpress.com/2010/05/20/install-json-c-in-linux/>

# KeePass

## Installation de KeePass sous Windows

Pour mieux vos nombreux mots de passe.

# LibXML2

Pour mieux manipuler vos documents XML.

## Récupération du contenu d’un nœud élément

Cela suppose que vous avez essayé de récupérer le contenu d’un nœud élément par la méthode (node->content) sans succès. Un nœud contient un nœud text. La meilleure façon de récupérer un nœud est de passer par la méthode (xmlNodeGetContent).

# OpenSSL

## Installation d’OpenSSL sous Windows

Pour accéder à une communication réseau mieux sécurisée.

Télécharger OpenSSL. Installer OpenSSL.

**WEB >** [**https://slproweb.com/products/Win32OpenSSL.html**](https://slproweb.com/products/Win32OpenSSL.html)

**Download Win32/Win64 OpenSSL**

**EXE**

# RaspberryPi

## Création d’un serveur de SMS

Pour pouvoir envoyer des SMS à partir de votre application. Cela suppose que vous disposez d’une clé WIFI avec intégration de carte SIM. Vous êtes déjà abonné chez un opérateur mobile.

## Liens

RaspiSMS ou comment faire un serveur SMS pour son installation domotique

<https://miniprojets.net/index.php/2020/04/02/raspisms-ou-comment-faire-un-serveur-sms-pour-son-installation-domotique/>

<https://documentation.raspisms.fr/developpers/api/overview.html>

<https://raspisms.fr/download/>

Jeedom ou comment domotiser sa maison en quelques clics

<https://miniprojets.net/index.php/2019/05/13/jeedom-ou-comment-domotiser-sa-maison-en-quelques-clics/>

Créer son premier capteur de température et envoyer les données via Lora

<https://miniprojets.net/index.php/2019/05/20/creer-son-premier-capteur-de-temperature-et-via-lora/>

# Sablotron

## Liens

Guide et référence Sablotron

<http://manual.freeshell.org/sablotron/sablot/apidocs/complete.html>

# SSL for Free

## Obtention d’un certificat SSL

Pour mieux sécuriser vos communications sur le réseau. Cela suppose que vous disposez d’un serveur dont les communications ne sont pas cryptées. Cela le rend vulnérable et moins sécurisé face à un intermédiaire malveillant qui écoute vos communications. Vous voulez les rendre plus sûres et plus sécurisées. Cela exige nécessairement l’utilisation d’un certificat SSL/TLS.

Obtenir un certificat SSL. Ouvrir un navigateur web. Ouvrir la page SSL for Free. Saisir le nom de domaine du site. Saisir l’adresse email et le mot de passe utilisateur.

**WEB >** [**https://www.sslforfree.com/**](https://www.sslforfree.com/)

**Secure > https > readydev.ovh > Create Free SSL Certificate**

**Domain > readydev.ovh**

**Email >** [**tiakagerard@hotmail.com**](mailto:tiakagerard@hotmail.com)

**Password > sslxxxxxx**

**Register**

**Domains > readydev.ovh >** [**www.readydev.ovh**](http://www.readydev.ovh)

**Validity > 90-Day Certificate**

**CSR & Contact > Cocher > Auto-Generate CSR**

**Finalize Your Order > Free**

**Next Step**

**Verification Method for readydev.ovh > http Upload File**

**> Téléchargez votre fichier d'authentification en utilisant le lien suivant : Télécharger le fichier d'authentification**

**> Téléchargez le fichier d'authentification sur votre serveur HTTP sous :** /.well-known/pki-validation/

**> Assurez-vous que votre fichier est disponible sous le lien suivant :** [**http://readydev.ovh/.well-known/pki-validation/1C2F09142096FEF67C4C34EBE1D0029D.txt**](http://readydev.ovh/.well-known/pki-validation/1C2F09142096FEF67C4C34EBE1D0029D.txt)

**> Assurez-vous que votre fichier est également disponible sous le lien suivant :** [**http://www.readydev.ovh/.well-known/pki-validation/1C2F09142096FEF67C4C34EBE1D0029D.txt**](http://www.readydev.ovh/.well-known/pki-validation/1C2F09142096FEF67C4C34EBE1D0029D.txt)

**> Cliquez sur "Étape suivante" pour continuer.**

**Finalize > Verify Domain**

**Download Certificate**

**Server Type > Default Format > Download certificate (.zip)**

**ZEROSSL > Dashboard > Issued**

**Readydev.ovh > Issued > Mar 28, 2023**

**Install > Revoke Certificate**

**Reason > A different certificate has been issued of this one**

**Cocher > I hereby confirm that i have read, understood and accepted the revocation terms**

**Revocate Certificate**

# ToirtoiseGit

Pour mieux gérer les différentes versions de votre projet de manière décentralisée à partir d’une interface utilisateur. Cela suppose que vous en avez assez d’utiliser Git en ligne de commande.

## Changement de branche

Cela suppose que vous ayez plusieurs branches dans votre référentiel git. Vous voulez basculer d’une branche à l’autre.

Ouvrir le menu contextuel.

**DEPOT\_LOCAL > clic droit**

Changer de branche.

**TortoiseGit > Switch/Checkout**

**Switch To > Branch > remote/origin/2.0**

**Options > Cocher > Create New Branch > 2.0**

**Cocher > Track**

**OK > Close**

## Suppression d’un fichier

Cela suppose que le fichier est déjà suivi dans git, sinon supprimer le simplement comme vous le faîtes habituellement.

Supprimer le fichier complêtement.

**mon\_fichier.txt > clic droit**

**TortoiseGit > Delete > OK**

Supprimer le fichier mais le garder en local.

**mon\_fichier.txt > clic droit**

**TortoiseGit > Delete (keep local) > OK**

## Sauvegarde des fichiers modifiés non validés

Cela suppose que vous avez terminé vos modifications et que vous souhaitez exporter vos modifications vers une machine distante en vue de réaliser des tests. Vous avez besoin d’exporter vos fichiers modifiés vers un répertoire local tout en gardant leur chemin complet dans le référentiel avant de les transférer vers la mchine distante. Cela vous évite de tranférer tout le dépôt local, car cela n’est pas nécessaire, et de ne transférer uniquement que les fichiers modifiés.

Ouvrir la fiche des validations. Rafraichir la fiche. Sélectionner les modifications. Sélection une destination. Exporter les modifications.

**TortoiseGit > Commit**

**F5 > pour actualiser**

**Ctrl + A > pour tout sélectionner**

**Clic droit > Export selection to**

**Dossier > tmp**

**Sélectionner un dossier**

# Ubuntu

## Modification des permissions d’un fichier

Cela suppose que vous avez un fichier dont les permissions vous empêchent d’effectuer cette opération.

Modifier les permissions d’un fichier. Modifier les permissions d’un dossier. Modifier les permissions d’un dossier en mode récursif.

**sudo chmod 777 mon\_fichier.txt**

**sudo chmod 777 mon\_dossier**

**sudo chmod -R 777 mon\_dossier**

## Modification du propriétaire et du groupe d’un fichier

Cela suppose que vous avez un fichier dont le propriétaire et le groupe d’appartenance sont incorrects. Cela peut vous empêcher d’effectuer cette opération sur le fichier.

Modifier le propriétaire et le groupe d’un fichier. Modifier le propriétaire et le groupe d’un dossier. Modifier le propriétaire et le groupe d’un dossier en mode récursif.

**sudo chown gkesse:gkesse mon\_fichier.txt**

**sudo chown gkesse:gkesse mon\_dossier**

**sudo chown -R gkesse:gkesse mon\_dossier**

## Création d’un nouveau screen

Cela suppose que vous disposez d’un serveur que vous relancer à chaque de nouvelle connexion SSH. Vous en avez assez de devoir le refaire à chaque fois. Vous voulez desormais lancer le serveur toujours en SSH en indiquant au système que son parent n’est pas le terminal de lancement, mais plus tôt le système lui-même, ainsi tant que le système n’est pas éteint le serveur restera toujours en activé même le terminal de lancement venait à être fermé.

Lancer un nouveau screen.

**screen -S gke**

## Liste des screen en cours d’exécution

Cela suppose que vous voulez accéder à un screen parmi plusieurs déjà créés.

Visualiser tous les screen en cours d’exécution.

**screen -ls**

## Accès à un screen déjà créé

Cela suppose que vous voulez modifier le comportement d’un programme tournant dans un screen déjà créé.

Accéder un screen existant.

**screen –r gke**

## Fermeture d’un screen déjà créé

Cela suppose que vous ne voulez plus faire tourner votre programme dans le screen.

Accéder au screen. Accéder au serveur. Fermer le screen.

**screen –r gke**

**exit**

# WampServer

Cela suppose que vous voulez lancer dans le développement d’applications web sous Windows.

## Installation de WampServer

Cela suppose que vous voulez lancer dans le développement d’applications web sous Windows.

Télécharger WampServer.

[**https://wampserver.aviatechno.net/**](https://wampserver.aviatechno.net/)

**Wampserver 3.2.6 64 bit x64**

**Wampserver update 3.2.9**

Installer WampServer.

**wampserver3.2.6\_x64.exe -> administrateur**

**langue -> francais**

**cocher -> je comprends et j accepte les termes du contrat de licence**

**suivant**

**suivant**

**localisation -> c:\wamp64**

**suivant**

**php -> 7.3.33**

**mariadb -> 10.6.5**

**mysql -> 8.0.27**

**suivant**

**menu -> Wampserver64**

**suivant**

**installer**

**navigateur -> chrome.exe**

**editeur -> notepad++**

**suivant**

**terminer**

Mettre à jour WampServer.

**wampserver3\_x86\_x64\_update3.2.9.exe**

## Démarrage de WampServer

Cela suppose que WampServer n’est pas encore lancé.

Ouvrir la barre de recherche Windows. Rechercher WampServer. Démarrer WampServer.

**WINDOWS > Bouton Windows**

**Barre de recherche > WampServer**

**WampServer > Oui > Icône vert**

Vérifier l’installation de WampServer.

[**http://localhost/**](http://localhost/)

## Redémarrage des services de WampServer

Cela suppose que vous avez déjà lancé WampServer. Vous avez fait des modifications dans les fichiers de configuration de WampServer. Vous devez redémarrer WampServer pour la prise en compte de vos modifications.

Redémarrer les services de WampServer.

**WampServer > clic gauche**

**Redémarrer les services**

## Configuration d’une machine virtuelle dans WampServer

Cela suppose que vous avez un répertoire local sur votre machine contenant les sources de votre serveur web et que vous voulez le rendre accessible via le navigateur.

Configurer le port du serveur. On gardera le port 80 pour l’accès au tableau de bord de WampServer.

**C:\wamp64\bin\apache\apache2.4.51\conf\httpd.conf**

**Listen 0.0.0.0:80**

**Listen [::0]:80**

**Listen 0.0.0.0:8001**

**Listen [::0]:8001**

Activer le module des machines virtuelles. Décommenter la ligne de chargement du module.

**C:\wamp64\bin\apache\apache2.4.51\conf\httpd.conf**

**LoadModule vhost\_alias\_module modules/mod\_vhost\_alias.so**

Indiquer le fichier de configuration des machines virtuelles. Décommenter la ligne d’inclusion du fichier.

**C:\wamp64\bin\apache\apache2.4.51\conf\httpd.conf**

**Include conf/extra/httpd-vhosts.conf**

Configurer la machine virtuelle du tableau de brod de WampServer.

**C:\wamp64\bin\apache\apache2.4.51\conf\extra\httpd-vhosts.conf**

**<VirtualHost \*:80>**

**ServerName localhost**

**ServerAlias localhost**

**DocumentRoot "${INSTALL\_DIR}/www"**

**<Directory "${INSTALL\_DIR}/www/">**

**Options +Indexes +Includes +FollowSymLinks +MultiViews**

**AllowOverride All**

**Require local**

**</Directory>**

**</VirtualHost>**

Configurer la machine virtuelle de votre serveur web.

**C:\wamp64\bin\apache\apache2.4.51\conf\extra\httpd-vhosts.conf**

**<VirtualHost \*:8001>**

**ServerName mon.serveur.com**

**ServerAlias www.mon.serveur.com**

**ServerAdmin mon.email@serveur.com**

**DocumentRoot C:\mon\chemin\web**

**ErrorLog C:\mon\chemin\web\logs\error.log**

**CustomLog C:\mon\chemin\web\logs\access.log combined**

**SetEnv MA\_VAR\_ENV "MA\_VALEUR"**

**<Directory C:\mon\chemin\web>**

**Options All**

**AllowOverride All**

**Order allow,deny**

**allow from all**

**Require all granted**

**</Directory>**

**</VirtualHost>**

Créer votre premier message sur votre serveur web. Cela suppose que vous êtes débutant.

**C:\mon\chemin\web\index.php**

<h1>**Bonjour tout le monde.**</h1>

Redémarrer les services de WampServer.

**WampServer > clic gauche**

**Redémarrer les services**

Vérifier la configuration de WampServer.

[**http://localhost:8001/**](http://localhost:8001/)

# Word 2007

## Activation du mode définir la langue

Cela suppose que vous avez des sections dans votre document qui sont formatées dans une langue dont vous n’en voulez pas. Vous avez peut-être des sections dans votre document qui n’ont pas besoin de vérification orthographique.

Activer le mode définir la langue. Ouvrir la fiche personnaliser la barre d’accès rapide. Ajouter le mode définir la langue.

**WORD > Bouton Office > Options Word**

**Personnaliser > Toutes les commandes**

**Définir la langue > Ajouter**

**OK**

# XSLT

## Liens

Tutoriel libxslt

<http://xmlsoft.org/xslt/tutorial/libxslttutorial.html>

Manuel de référence libxslt

<https://gnome.pages.gitlab.gnome.org/libxslt/devhelp/index.html>

Qu'est-ce que XSLT et à quoi sert-il ? Découvrez comment ça marche

<https://www.deepspace6.net/contribute/ds6-architecture.html>

# Fin