Сервис коротких ссылок / Задание

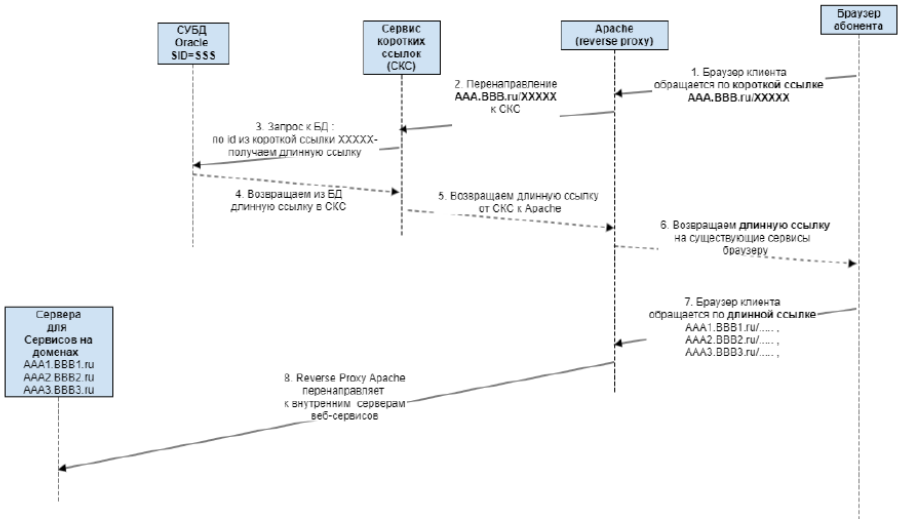
Задание 1: Разработать сервис сокращения ссылок URL. Длина ссылки должна содержать не более 13 символов. Аналог: https://clck.ru/

Задание 2: Разработать сервис перенаправления URL используя Reverse proxy в соотвествии с предоставленной схемой инфопотоков.

Сценарий: Клиент имеет короткую ссылку. Открывает эту ссылку в браузере. Происходит переаресация на длинный URL.

Инструментарий: Java, Oracle, Spring, Maven, Git.

Схема инфопотоков:



РЕШЕНИЕ

В архиве лежит 3 проекта:
server1 и server2 – обычные web-проекты, написанные на Java + Spring Web + Thymeleaf. Имеют по 2 эндпоинта с шаблонами, к которым мы бы редиректились через прокси сервер (короткая ссылка -> прокси -> short-clicker GET /api/v1 [api сервиса коротких ссылок] -> редирект через прокси).

Short-clicker – сервис коротких ссылок, написанный на Java + Spring Data + Spring Web + Thymeleaf. В нем содержатся 2 контроллера:

1. ClickerController – контроллер, предоставляющий шаблон с post-формой для генерации коротких ссылок и их добавления в БД PostgreSql.
2. ClickerControllerApi – контроллер, реализующий REST API сервиса. При обращении к эндпоинту GET /api/v1, с передаваемой короткой ссылкой (@RequestParam ‘url’), пользователь получал бы длинную ссылку, если такая короткая ссылка существует и между короткой ссылкой и длинной есть связь по внешнему ключу в БД.

Настройка Apache Reverse proxy (через XAMMP Control Panel):

1. В файле конфигурации сервера httpd.conf мы подключали 2 модуля для реализации проксирующего сервера: mod_proxy и mod_proxy_http + меняли порты в 2 переменных для успешного запуска сервера (Listen 8090 и ServerName localhost 8090).
2. В файле конфигурации сертификатов сервера httpd-ssl.conf мы изменяли порты в переменных (<VirtualHost _default_:4433> и ServerName localhost:4433) для предотвращения ошибок, связанных с резервированием портов.
3. Далее в конфиге httpd-vhosts.conf реализуем проксирование между сервисами (как обращаться по api к серверу и без костылей напрямую динамически изменять значения ProxyPass и ProxyPassReserve – я не нашел в официальной документации):
<VirtualHost *:*>
ProxyPreserveHost On

ProxyPass /RNjVxMolBb http://localhost:8080/api/v1?url=%2FRNjVxMolBb
ProxyPreserveHost /RNjVxMolBb http://localhost:8080/api/v1?url=%2FRNjVxMolBb

ProxyPass /yz4efsnTWl http://localhost:8080/api/v1?url=%2Fyz4efsnTWl
ProxyPreserveHost /yz4efsnTWl http://localhost:8080/api/v1?url=%2Fyz4efsnTWl

ProxyPass /R2gbp8dXia http://localhost:8080/api/v1?url=%2FR2gbp8dXia
ProxyPreserveHost /R2gbp8dXia http://localhost:8080/api/v1?url=%2FR2gbp8dXia

ProxyPass /n8l7NNByRJ http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ
ProxyPreserveHost /n8l7NNByRJ http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ

ProxyPass http://localhost:8080/api/v1?url=%2FRNjVxMolBb http://localhost:8081/firstLongURLFromServer1
ProxyPreserveHost http://localhost:8080/api/v1?url=%2FRNjVxMolBb http://localhost:8081/firstLongURLFromServer1

ProxyPass http://localhost:8080/api/v1?url=%2Fyz4efsnTWl http://localhost:8081/secondLongURLFromServer1
ProxyPreserveHost http://localhost:8080/api/v1?url=%2Fyz4efsnTWl http://localhost:8081/secondLongURLFromServer1

ProxyPass http://localhost:8080/api/v1?url=%2FR2gbp8dXia http://localhost:8081/firstLongURLFromServer2
ProxyPreserveHost http://localhost:8080/api/v1?url=%2FR2gbp8dXia http://localhost:8081/firstLongURLFromServer2

ProxyPass http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ http://localhost:8081/secondLongURLFromServer2
ProxyPreserveHost http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ http://localhost:8081/secondLongURLFromServer2

ServerName localhost
</VirtualHost>

КОДЫ СЕРВИСОВ И ОПИСАНИЕ БД POSTGRESQL

<u>База данных</u>

longurls × shorturls ClickerController.java

4 rows

WHERE ORDER BY

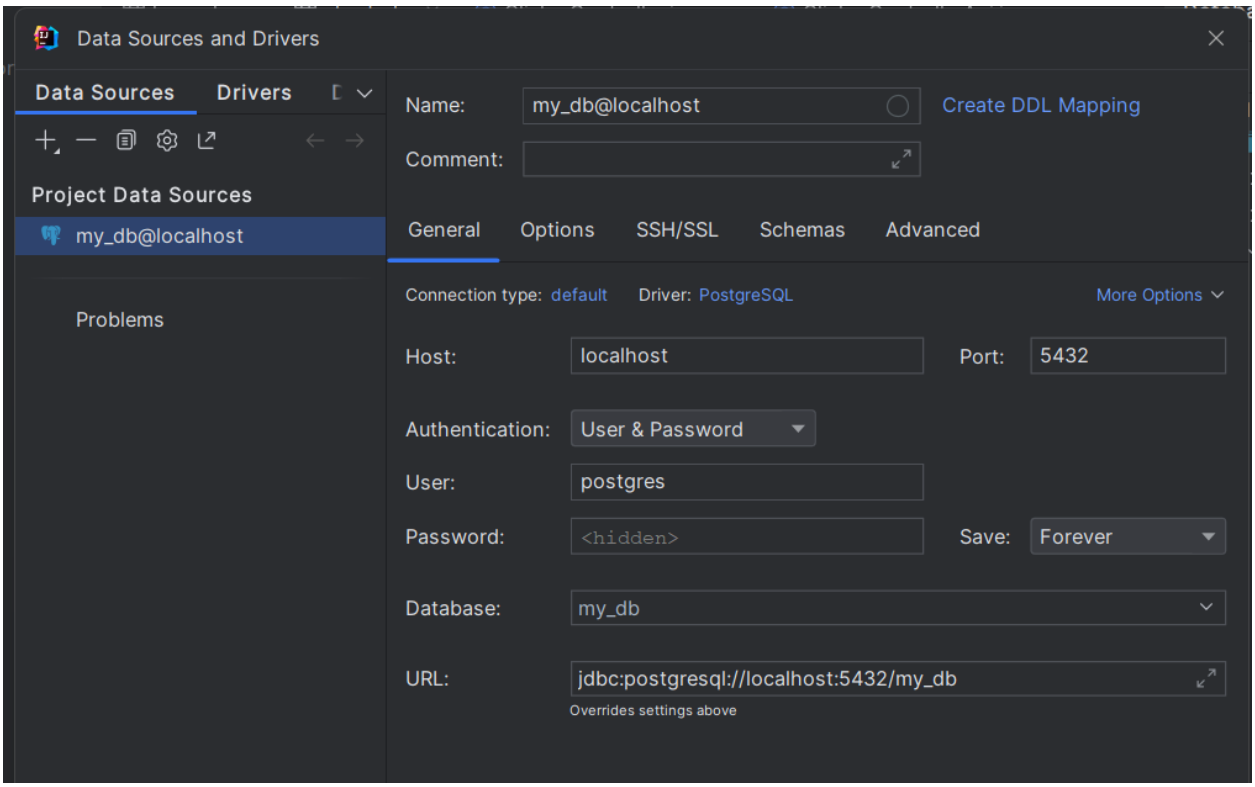
	id	url
1	1	/firstLongURLFromServer1
2	2	/secondLongURLFromServer1
3	3	/firstLongURLFromServer2
4	4	/secondLongURLFromServer2

longurls × shorturls × ClickerController.java

4 rows Tx:

WHERE ORDER BY

	id	url	longurl_id
1	1	/RNjVxMolBb	1
2	2	/yz4efsnTWl	2
3	3	/R2gbp8dXia	3
4	4	/n8l7NNByRJ	4


Data Sources and Drivers
Data Sources
Drivers
Project Data Sources
my_db@localhost
Problems
Name:
my_db@localhost
Create DDL Mapping
Comment:
General
Options
SSH/SSL
Schemas
Advanced
Connection type: default
Driver: PostgreSQL
More Options
Host:
localhost
Port:
5432
Authentication:
User & Password
User:
postgres
Password:
<hidden>
Save:
Forever
Database:
my_db
URL:
jdbc:postgresql://localhost:5432/my_db
Overrides settings above

Server1

MyController.java

```
package com.example.server1.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class MyController {
   @RequestMapping(value = "/firstLongURLFromServer1", method =
RequestMethod.GET)
   @ResponseBody
   public String first() {
      return "Server1, firstLongURLFromServer1";
   }

   @RequestMapping(value = "/secondLongURLFromServer1", method =
RequestMethod.GET)
   @ResponseBody
   public String second() {
      return "Server1, secondLongURLFromServer1";
   }
}
```

Application.properties

```
server.port=8081
```

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>3.0.5</version>
                <relativePath/> <!-- lookup parent from repository -->
```

```
        </parent>
        <groupId>com.example</groupId>
        <artifactId>server1</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <name>server1</name>
        <description>Demo project for Spring Boot</description>
        <properties>
            <java.version>17</java.version>
        </properties>
        <dependencies>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-thymeleaf</artifactId>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
            </dependency>

            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
            </dependency>
        </dependencies>

        <build>
            <plugins>
                <plugin>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-maven-plugin</artifactId>
                </plugin>
            </plugins>
        </build>

    </project>
```

<u>Server2</u>

<u>MyController.java</u>

```
package com.example.server2.controller;

import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class MyController {
   @RequestMapping(value = "/firstLongURLFromServer2", method =
RequestMethod.GET)
   @ResponseBody
   public String first() {
      return "Server2, firstLongURLFromServer2";
   }

   @RequestMapping(value = "/secondLongURLFromServer2", method =
RequestMethod.GET)
   @ResponseBody
   public String second() {
      return "Server2, secondLongURLFromServer2";
   }
}
```

Application.properties

```
server.port=8083
```

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>3.0.5</version>
                <relativePath/> <!-- lookup parent from repository -->
        </parent>
        <groupId>com.example</groupId>
        <artifactId>server2</artifactId>
        <version>0.0.1-SNAPSHOT</version>
```

```
        <name>server2</name>
        <description>Demo project for Spring Boot</description>
        <properties>
            <java.version>17</java.version>
        </properties>
        <dependencies>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-thymeleaf</artifactId>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
            </dependency>

            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
            </dependency>
        </dependencies>

        <build>
            <plugins>
                <plugin>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-maven-plugin</artifactId>
                </plugin>
            </plugins>
        </build>

    </project>
```

Short-clicker

ClickerController.java

```
package com.example.shortclicker.controller;

import com.example.shortclicker.database.LongUrls;
import com.example.shortclicker.database.LongUrlsRepository;
import com.example.shortclicker.database.ShortUrls;
import com.example.shortclicker.database.ShortUrlsRepository;
import com.example.shortclicker.utility.UrlSubmit;
import org.apache.commons.lang3.RandomStringUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class ClickerController {
   @Autowired
   LongUrlsRepository longUrlsRepository;

   @Autowired
   ShortUrlsRepository shortUrlsRepository;

   @GetMapping("/main")
   public String main(Model model) {
      model.addAttribute("urlSubmit", new UrlSubmit());
      return "main";
   }

   @PostMapping("/main")
   public String mainSubmit(@ModelAttribute UrlSubmit urlSubmit, Model
model) {
      LongUrls longUrlObj =
longUrlsRepository.findLongUrlsByUrl(urlSubmit.getUrl());
      ShortUrls shortUrls = null;
      if (longUrlObj != null) {
         shortUrls = new ShortUrls();
         shortUrls.setUrl("/" + RandomStringUtils.random(10, true, true));
         shortUrls.setLongUrlId(longUrlObj.getId());
```

```
            ShortUrls lastShortUrl =
shortUrlsRepository.findFirstByOrderByIdDesc();
            shortUrls.setId(lastShortUrl.getId() + 1);
            shortUrlsRepository.save(shortUrls);
        }
        model.addAttribute("urlSubmit", urlSubmit);
        model.addAttribute("shortUrl", shortUrls);
        return "main";
    }
}
```

<u>Шаблон main.html (для ClickerController.java)</u>

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Главная</title>
</head>
<body>
<form action="#" th:action="@{/main}" th:object="${urlSubmit}"
method="post">
  <p>Введите ссылку:</p>
  <label title="Введите ссылку">
    <input type="text" th:field="*{url}">
  </label>
  <th:block th:if="${shortUrl} == null">
      <input type="submit">
  </th:block>
    <th:block th:unless="${shortUrl} == null">
        <input type="submit" disabled>
    </th:block>
</form>
<th:block th:if="${shortUrl} != null">
    <p th:text="'Ваша ссылка: ' + ${shortUrl.getUrl()}"/>
    <form action="#" th:action="@{/main}" method="get">
        <input type="submit" value="Вернуться">
    </form>
</th:block>
</body>
</html>
```

ClickerControllerApi.java

```
package com.example.shortclicker.controller;

import com.example.shortclicker.database.LongUrls;
import com.example.shortclicker.database.LongUrlsRepository;
import com.example.shortclicker.database.ShortUrls;
import com.example.shortclicker.database.ShortUrlsRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

@CrossOrigin(origins = "http://localhost:8080")
@RestController
@RequestMapping("/api/v1")
public class ClickerControllerApi {
   @Autowired
   LongUrlsRepository longUrlsRepository;
   @Autowired
   ShortUrlsRepository shortUrlsRepository;

   @GetMapping
   public ResponseEntity<String> getLongUrl(@RequestParam(name =
"url") String url) {
      url = url.replace("https://", "");
      ShortUrls shortUrl = shortUrlsRepository.findShortUrlsByUrl(url);
      if (shortUrl != null) {
         LongUrls longUrl =
longUrlsRepository.findLongUrlsById(shortUrl.getLongUrlId());
         return new ResponseEntity<>(longUrl.getUrl(), HttpStatus.OK);
      }
      return new ResponseEntity<>("https://localhost:8080/main",
HttpStatus.NOT_FOUND);
   }
}
```

LongUrls.java (модель таблицы длинных ссылок из БД)

```
package com.example.shortclicker.database;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.Getter;
import lombok.RequiredArgsConstructor;
import lombok.Setter;

@Entity
@Table(name = "longurls")
@RequiredArgsConstructor
@Getter
@Setter
public class LongUrls {
    @Id
    private Integer id;

    @Column(name = "url")
    private String url;
}
```

ShortUrls.java (модель таблицы коротких ссылок из БД)

```
package com.example.shortclicker.database;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import lombok.Getter;
import lombok.RequiredArgsConstructor;
import lombok.Setter;

@Entity
@Table(name = "longurls")
@RequiredArgsConstructor
@Getter
@Setter
public class LongUrls {
    @Id
```

```
    private Integer id;

    @Column(name = "url")
    private String url;
}
```

<u>JPA-репозитории моделей для коротких и длинных ссылок:</u>

```
package com.example.shortclicker.database;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface LongUrlsRepository extends JpaRepository<LongUrls,
Integer> {
    LongUrls findLongUrlsByUrl(String url);
    LongUrls findLongUrlsById(Integer id);
}
-------------------------------------------------------
package com.example.shortclicker.database;

import org.springframework.data.jpa.repository.JpaRepository;

public interface ShortUrlsRepository extends JpaRepository<ShortUrls,
Integer> {
    ShortUrls findFirstByOrderByIdDesc();
    ShortUrls findShortUrlsByUrl(String url);
}
```

<u>Класс, выполняющий функционал ModelAttribute, для получения данных из POST-формы:</u>

```
package com.example.shortclicker.utility;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
public class UrlSubmit {
    private String url;
}
```

Application.properties

```
server.port=8080

spring.main.banner-mode=off
logging.level.org.springframework=ERROR

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

spring.sql.init.mode=never
spring.sql.init.platform=postgres
spring.datasource.driver-class-name=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5432/my_db
spring.datasource.username=postgres
spring.datasource.password=123

spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation=true
```

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>3.0.5</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.example</groupId>
	<artifactId>short-clicker</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>short-clicker</name>
	<description>Demo project for Spring Boot</description>
	<properties>
		<java.version>17</java.version>
	</properties>
	<dependencies>
		<dependency>
```

```
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.postgresql</groupId>
            <artifactId>postgresql</artifactId>
            <version>42.5.0</version>
        </dependency>

        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
        </dependency>

        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-lang3</artifactId>
            <version>3.12.0</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
```

```
            </plugins>
        </build>

</project>
```

<u>Конфиг httpd.conf:</u>

```
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/access_log"
# with ServerRoot set to "/usr/local/apache2" will be interpreted by the
# server as "/usr/local/apache2/logs/access_log", whereas "/logs/access_log"
# will be interpreted as '/logs/access_log'.
#
# NOTE: Where filenames are specified, you must use forward slashes
# instead of backslashes (e.g., "c:/apache" instead of "c:\apache").
# If a drive letter is omitted, the drive on which httpd.exe is located
# will be used by default.  It is recommended that you always supply
# an explicit drive letter in absolute paths to avoid confusion.

#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
```

```
Define SRVROOT "C:/Program Files/xammp/apache"

ServerRoot "C:/Program Files/xammp/apache"

#
# Mutex: Allows you to set the mutex mechanism and mutex file directory
# for individual mutexes, or change the global defaults
#
# Uncomment and change the directory if mutexes are file-based and the
default
# mutex file directory is not on a local disk or is not appropriate for some
# other reason.
#
# Mutex default:logs

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO
you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
```

```
LoadModule allowmethods_module modules/mod_allowmethods.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule auth_form_module modules/mod_auth_form.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
LoadModule authn_core_module modules/mod_authn_core.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authn_socache_module modules/mod_authn_socache.so
#LoadModule authnz_fcgi_module modules/mod_authnz_fcgi.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule authz_core_module modules/mod_authz_core.so
#LoadModule authz_dbd_module modules/mod_authz_dbd.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
#LoadModule authz_owner_module modules/mod_authz_owner.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule brotli_module modules/mod_brotli.so
#LoadModule buffer_module modules/mod_buffer.so
#LoadModule cache_module modules/mod_cache.so
#LoadModule cache_disk_module modules/mod_cache_disk.so
#LoadModule cache_socache_module modules/mod_cache_socache.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule charset_lite_module modules/mod_charset_lite.so
#LoadModule data_module modules/mod_data.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule dav_lock_module modules/mod_dav_lock.so
#LoadModule dbd_module modules/mod_dbd.so
#LoadModule deflate_module modules/mod_deflate.so
LoadModule dir_module modules/mod_dir.so
#LoadModule dumpio_module modules/mod_dumpio.so
LoadModule env_module modules/mod_env.so
#LoadModule expires_module modules/mod_expires.so
#LoadModule ext_filter_module modules/mod_ext_filter.so
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule filter_module modules/mod_filter.so
```

```
#LoadModule http2_module modules/mod_http2.so
LoadModule headers_module modules/mod_headers.so
#LoadModule heartbeat_module modules/mod_heartbeat.so
#LoadModule heartmonitor_module modules/mod_heartmonitor.so
#LoadModule ident_module modules/mod_ident.so
#LoadModule imagemap_module modules/mod_imagemap.so
LoadModule include_module modules/mod_include.so
LoadModule info_module modules/mod_info.so
LoadModule isapi_module modules/mod_isapi.so
#LoadModule lbmethod_bybusyness_module
modules/mod_lbmethod_bybusyness.so
#LoadModule lbmethod_byrequests_module
modules/mod_lbmethod_byrequests.so
#LoadModule lbmethod_bytraffic_module
modules/mod_lbmethod_bytraffic.so
#LoadModule lbmethod_heartbeat_module
modules/mod_lbmethod_heartbeat.so
#LoadModule ldap_module modules/mod_ldap.so
#LoadModule logio_module modules/mod_logio.so
LoadModule log_config_module modules/mod_log_config.so
#LoadModule log_debug_module modules/mod_log_debug.so
#LoadModule log_forensic_module modules/mod_log_forensic.so
#LoadModule lua_module modules/mod_lua.so
LoadModule cache_disk_module modules/mod_cache_disk.so
#LoadModule macro_module modules/mod_macro.so
#LoadModule md_module modules/mod_md.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_express_module modules/mod_proxy_express.so
#LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_hcheck_module modules/mod_proxy_hcheck.so
#LoadModule proxy_html_module modules/mod_proxy_html.so
LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_http2_module modules/mod_proxy_http2.so
#LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
#LoadModule proxy_uwsgi_module modules/mod_proxy_uwsgi.so
```

```
#LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
#LoadModule ratelimit_module modules/mod_ratelimit.so
#LoadModule reflector_module modules/mod_reflector.so
#LoadModule remoteip_module modules/mod_remoteip.so
#LoadModule request_module modules/mod_request.so
#LoadModule reqtimeout_module modules/mod_reqtimeout.so
LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule sed_module modules/mod_sed.so
#LoadModule session_module modules/mod_session.so
#LoadModule session_cookie_module modules/mod_session_cookie.so
#LoadModule session_crypto_module modules/mod_session_crypto.so
#LoadModule session_dbd_module modules/mod_session_dbd.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule slotmem_plain_module modules/mod_slotmem_plain.so
#LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
#LoadModule socache_dbm_module modules/mod_socache_dbm.so
#LoadModule socache_memcache_module
modules/mod_socache_memcache.so
#LoadModule socache_redis_module modules/mod_socache_redis.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
#LoadModule speling_module modules/mod_speling.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule status_module modules/mod_status.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule unique_id_module modules/mod_unique_id.so
#LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
LoadModule version_module modules/mod_version.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
#LoadModule watchdog_module modules/mod_watchdog.so
#LoadModule xml2enc_module modules/mod_xml2enc.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User daemon
```

```
Group daemon

</IfModule>

# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition.  These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#

#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#
ServerAdmin postmaster@localhost

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you
specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:80

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>
```

```
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/Program Files/xammp/htdocs"
<Directory "C:/Program Files/xammp/htdocs">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI
MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks Includes ExecCGI

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   AllowOverride FileInfo AuthConfig Limit
    #
    AllowOverride All

    #
    # Controls who can get stuff from this server.
    #
    Require all granted
</Directory>
```

```
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.php index.pl index.cgi index.asp index.shtml
index.html index.htm \
                default.php default.pl default.cgi default.asp default.shtml
default.html default.htm \
                home.php home.pl home.cgi home.asp home.shtml home.html
home.htm
</IfModule>

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
    Require all denied
</Files>

#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog "logs/error.log"

#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn

<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
```

```
    #
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    #CustomLog "logs/access.log" common

    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #
    CustomLog "logs/access.log" combined
</IfModule>

<IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    # exist in your server's namespace, but do not anymore. The client
    # will make a new request for the document at its new location.
    # Example:
    # Redirect permanent /foo http://www.example.com/bar

    #
    # Alias: Maps web paths into filesystem paths and is used to
    # access content that does not live under the DocumentRoot.
    # Example:
    # Alias /webpath /full/filesystem/path
    #
    # If you include a trailing / on /webpath then the server will
```

```
    # require it to be present in the URL.  You will also likely
    # need to provide a <Directory> section to allow access to
    # the filesystem path.

    #
    # ScriptAlias: This controls which directories contain server scripts.
    # ScriptAliases are essentially the same as Aliases, except that
    # documents in the target directory are treated as applications and
    # run by the server when requested rather than as documents sent to the
    # client.  The same rules about trailing "/" apply to ScriptAlias
    # directives as to Alias.
    #
    ScriptAlias /cgi-bin/ "C:/Program Files/xammp/cgi-bin/"

</IfModule>

<IfModule cgid_module>
    #
    # ScriptSock: On threaded servers, designate the path to the UNIX
    # socket used to communicate with the CGI daemon of mod_cgid.
    #
    #Scriptsock cgisock
</IfModule>

#
# "C:/Program Files/xammp/cgi-bin" should be changed to whatever your
ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "C:/Program Files/xammp/cgi-bin">
    AllowOverride All
    Options None
    Require all granted
</Directory>

<IfModule headers_module>
    #
    # Avoid passing HTTP_PROXY environment to CGI's on this or any
proxied
    # backend servers which have lingering "httpoxy" defects.
    # 'Proxy' request header is undefined by the IETF, not listed by IANA
    #
```

```
    RequestHeader unset Proxy early
</IfModule>

<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    #
    TypesConfig conf/mime.types

    #
    # AddType allows you to add to or override the MIME configuration
    # file specified in TypesConfig for specific file types.
    #
    #AddType application/x-gzip .tgz
    #
    # AddEncoding allows you to have certain browsers uncompress
    # information on the fly. Note: Not all browsers support this.
    #
    #AddEncoding x-compress .Z
    #AddEncoding x-gzip .gz .tgz
    #
    # If the AddEncoding directives above are commented-out, then you
    # probably should define those extensions to indicate media types:
    #
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz

    #
    # AddHandler allows you to map certain file extensions to "handlers":
    # actions unrelated to filetype. These can be either built into the server
    # or added with the Action directive (see below)
    #
    # To use CGI scripts outside of ScriptAliased directories:
    # (You will also need to add "ExecCGI" to the "Options" directive.)
    #
    AddHandler cgi-script .cgi .pl .asp

    # For type maps (negotiated resources):
    #AddHandler type-map var

    #
```

```
    # Filters allow you to process content before it is sent to the client.
    #
    # To parse .shtml files for server-side includes (SSI):
    # (You will also need to add "Includes" to the "Options" directive.)
    #
    AddType text/html .shtml
    AddOutputFilter INCLUDES .shtml
</IfModule>

#
# The mod_mime_magic module allows the server to use various hints from
the
# contents of the file itself to determine its type.  The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
<IfModule mime_magic_module>
    #
    # The mod_mime_magic module allows the server to use various hints
from the
    # contents of the file itself to determine its type.  The MIMEMagicFile
    # directive tells the module where the hint definitions are located.
    #
    MIMEMagicFile "conf/magic"
</IfModule>


#
# Customizable error responses come in three flavors:
# 1) plain text 2) local redirects 3) external redirects
#
# Some examples:
#ErrorDocument 500 "The server made a boo boo."
#ErrorDocument 404 /missing.html
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
#ErrorDocument 402 http://www.example.com/subscription_info.html
#

#
# MaxRanges: Maximum number of Ranges in a request before
# returning the entire resource, or one of the special
# values 'default', 'none' or 'unlimited'.
# Default setting is to accept 200 Ranges.
```

```
#MaxRanges unlimited

#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files.  This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
#EnableSendfile off

# Supplemental configuration
#
# The configuration files in the conf/extra/ directory can be
# included to add extra features or to modify the default configuration of
# the server, or you may simply copy their contents here and change as
# necessary.

# Server-pool management (MPM specific)
Include conf/extra/httpd-mpm.conf

# Multi-language error messages
#Include conf/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
Include conf/extra/httpd-autoindex.conf

# Language settings
Include conf/extra/httpd-languages.conf

# User home directories
Include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
Include conf/extra/httpd-info.conf

# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

```
# Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf

# Distributed authoring and versioning (WebDAV)
#Attention! WEB_DAV is a security risk without a new userspecific
configuration for a secure authentifcation
#Include conf/extra/httpd-dav.conf

# Various default settings
#Include conf/extra/httpd-default.conf
# Implements a proxy/gateway for Apache.
Include "conf/extra/httpd-proxy.conf"
# Various default settings
Include "conf/extra/httpd-default.conf"
# XAMPP settings
Include "conf/extra/httpd-xampp.conf"

# Configure mod_proxy_html to understand HTML4/XHTML1
<IfModule proxy_html_module>
Include conf/extra/proxy-html.conf
</IfModule>

# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

# XAMPP: We disable operating system specific optimizations for a
listening
# socket by the http protocol here. IE 64 bit make problems without this.

AcceptFilter http none
AcceptFilter https none
# AJP13 Proxy
<IfModule mod_proxy.c>
```

```
<IfModule mod_proxy_ajp.c>
Include "conf/extra/httpd-ajp.conf"
</IfModule>
</IfModule>
```

Конфиг httpd-ssl.conf:

```
#
# This is the Apache server configuration file providing SSL support.
# It contains the configuration directives to instruct the server how to
# serve pages over an https connection. For detailed information about these
# directives see <URL:http://httpd.apache.org/docs/2.4/mod/mod_ssl.html>
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# Required modules: mod_log_config, mod_setenvif, mod_ssl,
#          socache_shmcb_module (for default value of SSLSessionCache)

#
# Pseudo Random Number Generator (PRNG):
# Configure one or more sources to seed the PRNG of the SSL library.
# The seed data should be of good random quality.
# WARNING! On some platforms /dev/random blocks if not enough entropy
# is available. This means you then cannot use the /dev/random device
# because it would lead to very long connection times (as long as
# it requires to make more entropy available). But usually those
# platforms additionally provide a /dev/urandom device which doesn't
# block. So, if available, use this one instead. Read the mod_ssl User
# Manual for more details.
#
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512


#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
Listen 443
```

```
##
##  SSL Global Context
##
##  All SSL configuration in this context applies both to
##  the main server and all SSL-enabled virtual hosts.
##

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate,
#   and that httpd will negotiate as the client of a proxied server.
#   See the OpenSSL documentation for a complete list of ciphers, and
#   ensure these follow appropriate best practices for this deployment.
#   httpd 2.2.30, 2.4.13 and later force-disable aNULL, eNULL and EXP
ciphers,
#   while OpenSSL disabled these by default in 0.9.8zf/1.0.0r/1.0.1m/1.0.2a.
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES
SSLProxyCipherSuite HIGH:MEDIUM:!MD5:!RC4:!3DES

#  By the end of 2016, only TLSv1.2 ciphers should remain in use.
#  Older ciphers should be disallowed as soon as possible, while the
#  kRSA ciphers do not offer forward secrecy.  These changes inhibit
#  older clients (such as IE6 SP2 or IE8 on Windows XP, or other legacy
#  non-browser tooling) from successfully connecting.
#
#  To restrict mod_ssl to use only TLSv1.2 ciphers, and disable
#  those protocols which do not support forward secrecy, replace
#  the SSLCipherSuite and SSLProxyCipherSuite directives above with
#  the following two directives, as soon as practical.
# SSLCipherSuite HIGH:MEDIUM:!SSLv3:!kRSA
# SSLProxyCipherSuite HIGH:MEDIUM:!SSLv3:!kRSA

#   User agents such as web browsers are not configured for the user's
#   own preference of either security or performance, therefore this
#   must be the prerogative of the web server administrator who manages
#   cpu load versus confidentiality, so enforce the server's cipher order.
SSLHonorCipherOrder on

#   SSL Protocol support:
#   List the protocol versions which clients are allowed to connect with.
#   Disable SSLv3 by default (cf. RFC 7525 3.1.1).  TLSv1 (1.0) should be
#   disabled as quickly as practical.  By the end of 2016, only the TLSv1.2
```

```
#   protocol or later should remain in use.
SSLProtocol all -SSLv3
SSLProxyProtocol all -SSLv3

#   Pass Phrase Dialog:
#   Configure the pass phrase gathering process.
#   The filtering dialog program (`builtin' is an internal
#   terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog  builtin

#   Inter-Process Session Cache:
#   Configure the SSL Session Cache: First the mechanism
#   to use and second the expiring timeout (in seconds).
#SSLSessionCache "shmcb:C:/Program
Files/xammp/apache/logs/ssl_scache(512000)"
SSLSessionCache "shmcb:C:/Program
Files/xammp/apache/logs/ssl_scache(512000)"
SSLSessionCacheTimeout  300

#   OCSP Stapling (requires OpenSSL 0.9.8h or later)
#
#   This feature is disabled by default and requires at least
#   the two directives SSLUseStapling and SSLStaplingCache.
#   Refer to the documentation on OCSP Stapling in the SSL/TLS
#   How-To for more information.
#
#   Enable stapling for all SSL-enabled servers:
#SSLUseStapling On

#   Define a relatively small cache for OCSP Stapling using
#   the same mechanism that is used for the SSL session cache
#   above.  If stapling is used with more than a few certificates,
#   the size may need to be increased.  (AH01929 will be logged.)
#SSLStaplingCache "shmcb:${SRVROOT}/logs/ssl_stapling(32768)"

#   Seconds before valid OCSP responses are expired from the cache
#SSLStaplingStandardCacheTimeout 3600

#   Seconds before invalid OCSP responses are expired from the cache
#SSLStaplingErrorCacheTimeout 600

##
```

```
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

#   General setup for the virtual host
DocumentRoot "C:/Program Files/xammp/htdocs"
ServerName www.example.com:4433
ServerAdmin admin@example.com
ErrorLog "C:/Program Files/xammp/apache/logs/error.log"
TransferLog "C:/Program Files/xammp/apache/logs/access.log"

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   Server Certificate:
#   Point SSLCertificateFile "conf/ssl.crt/server.crt"
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
#   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
#   require an ECC certificate which can also be configured in
#   parallel.
SSLCertificateFile "conf/ssl.crt/server.crt"
#SSLCertificateFile "conf/ssl.crt/server.crt"
#SSLCertificateFile "conf/ssl.crt/server.crt"

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile "conf/ssl.key/server.key"
#SSLCertificateKeyFile "conf/ssl.key/server.key"
#SSLCertificateKeyFile "conf/ssl.key/server.key"

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
```

```
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
"conf/ssl.crt/server.crt"
#   certificate for convenience.
#SSLCertificateChainFile "${SRVROOT}/conf/server-ca.crt"

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCACertificatePath "${SRVROOT}/conf/ssl.crt"
#SSLCACertificateFile "${SRVROOT}/conf/ssl.crt/ca-bundle.crt"

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded).
#   The CRL checking mode needs to be configured explicitly
#   through SSLCARevocationCheck (defaults to "none" otherwise).
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath "${SRVROOT}/conf/ssl.crl"
#SSLCARevocationFile "${SRVROOT}/conf/ssl.crl/ca-bundle.crl"
#SSLCARevocationCheck chain

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   TLS-SRP mutual authentication:
#   Enable TLS-SRP and set the path to the OpenSSL SRP verifier
#   file (containing login information for SRP user accounts).
#   Requires OpenSSL 1.0.1 or newer. See the mod_ssl FAQ for
```

```
#   detailed instructions on creating this file. Example:
#   "openssl srp -srpvfile ${SRVROOT}/conf/passwd.srpv -add username"
#SSLSRPVerifierFile "${SRVROOT}/conf/passwd.srpv"

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#           and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#           and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#           and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#           and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) \
#          or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.
The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables:
SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of
the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment
variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
```

```
#     exportation for CGI and SSI requests only.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied
even
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when
SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory "C:/Program Files/xammp/apache/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is sent or allowed to be received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach
where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
#     SSL close notify alert is send and mod_ssl waits for the close notify
#     alert of the client. This is 100% SSL/TLS standard compliant, but in
#     practice often causes hanging connections with brain-dead browsers.
Use
#     this only for browsers where you know that their SSL implementation
#     works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
```

```
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0"
and
#   "force-response-1.0" for this.
BrowserMatch "MSIE [2-5]" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0

#   Per-Server Logging:
#   The home of a custom SSL log file. Use this when you want a
#   compact non-error SSL logfile on a virtual host basis.
CustomLog "C:/Program Files/xammp/apache/logs/ssl_request.log" \
        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>
```

<u>Конфиг apache/conf/extra/httpd-vhosts.conf:</u>

```
# Virtual Hosts
#
# Required modules: mod_log_config

# If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them. Most
configurations
# use only name-based virtual hosts so the server doesn't need to worry
about
# IP addresses. This is indicated by the asterisks in the directives below.
#
# Please see the documentation at
# <URL:http://httpd.apache.org/docs/2.4/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-S' to verify your virtual host
# configuration.

#
# Use name-based virtual hosting.
#
##NameVirtualHost *:80
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
```

```
# match a ##ServerName or ##ServerAlias in any <VirtualHost> block.
#
##<VirtualHost *:80>
   ##ServerAdmin webmaster@dummy-host.example.com
   ##DocumentRoot "C:/Program Files/xammp/htdocs/dummy-
host.example.com"
   ##ServerName dummy-host.example.com
   ##ServerAlias www.dummy-host.example.com
   ##ErrorLog "logs/dummy-host.example.com-error.log"
   ##CustomLog "logs/dummy-host.example.com-access.log" common
##</VirtualHost>

##<VirtualHost *:80>
   ##ServerAdmin webmaster@dummy-host2.example.com
   ##DocumentRoot "C:/Program Files/xammp/htdocs/dummy-
host2.example.com"
   ##ServerName dummy-host2.example.com
   ##ErrorLog "logs/dummy-host2.example.com-error.log"
   ##CustomLog "logs/dummy-host2.example.com-access.log" common
##</VirtualHost>



<VirtualHost *:*>
   ProxyPreserveHost On

   ProxyPass /RNjVxMolBb
http://localhost:8080/api/v1?url=%2FRNjVxMolBb
   ProxyPreserveHost /RNjVxMolBb
http://localhost:8080/api/v1?url=%2FRNjVxMolBb

   ProxyPass /yz4efsnTWl
http://localhost:8080/api/v1?url=%2Fyz4efsnTWl
   ProxyPreserveHost /yz4efsnTWl
http://localhost:8080/api/v1?url=%2Fyz4efsnTWl

   ProxyPass /R2gbp8dXia
http://localhost:8080/api/v1?url=%2FR2gbp8dXia
   ProxyPreserveHost /R2gbp8dXia
http://localhost:8080/api/v1?url=%2FR2gbp8dXia
```

```
    ProxyPass /n8l7NNByRJ
http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ
    ProxyPreserveHost /n8l7NNByRJ
http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ

    ProxyPass http://localhost:8080/api/v1?url=%2FRNjVxMolBb
http://localhost:8081/firstLongURLFromServer1
    ProxyPreserveHost http://localhost:8080/api/v1?url=%2FRNjVxMolBb
http://localhost:8081/firstLongURLFromServer1

    ProxyPass http://localhost:8080/api/v1?url=%2Fyz4efsnTWl
http://localhost:8081/secondLongURLFromServer1
    ProxyPreserveHost http://localhost:8080/api/v1?url=%2Fyz4efsnTWl
http://localhost:8081/secondLongURLFromServer1

    ProxyPass http://localhost:8080/api/v1?url=%2FR2gbp8dXia
http://localhost:8081/firstLongURLFromServer2
    ProxyPreserveHost http://localhost:8080/api/v1?url=%2FR2gbp8dXia
http://localhost:8081/firstLongURLFromServer2

    ProxyPass http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ
http://localhost:8081/secondLongURLFromServer2
    ProxyPreserveHost http://localhost:8080/api/v1?url=%2Fn8l7NNByRJ
http://localhost:8081/secondLongURLFromServer2

    ServerName localhost
</VirtualHost>
```