Teknisk dokumentation - Community

# Inledning

Det här är den tekniska dokumentationen för community-delen av projektet. Här finns det lite korta tutorials där det står hur man ska gå tillväga när man exempelvis ska installera ett nytt plugin eller lägga till en helt ny sida. Det finns även en del tips på saker som är bra att känna till när man arbetar med wordpress, detta kan vara saker som funktioner eller var data sparas.

Genom att läsa igenom denna dokumentation borde du ha en bra förståelse för hur du ska komma igång både med wordpress och vårat API.

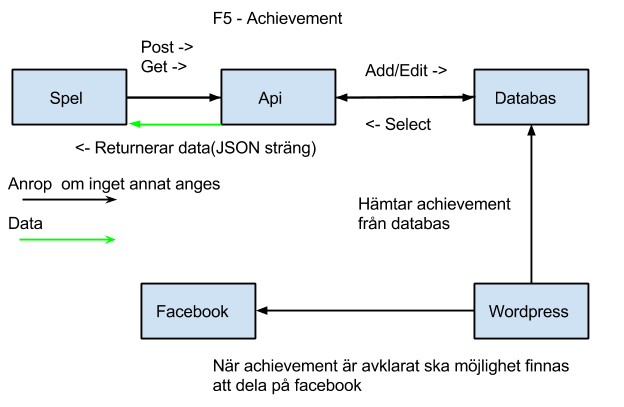
## Innehållsförtecking

Sida

|  |  |
| --- | --- |
| [Arkitektuellt signifikant krav](#h.pp9m5rt3j6bz) | 2 |
| [Hur får jag igång wordpress på min lokala utvecklingsmiljö](#h.qvlj3sqzev2p) | 3 |
| [Vilka api-er som används](#h.xoe1yv53skcf) | 3 |
| [Vilka plugins som används i wordpress](#h.r21jw2k133ix) | 3 |
| [Kodstruktur wordpress](#h.984aainnxhr3) | 3 |
| [Fördelar med wordpress](#h.cb46ukxl9vaj) | 3 |
| [Nackdelar med wordpress](#h.ozvric2hqg7x) | 3 |
| [Bra klasser, filer och funktioner att känna till innan du börjar](#h.hzo9xygl3zzu) | 4 |
| [Något som jag ska tänka på innan jag börjar?](#h.9kzjuqs7hpp0) | 4 |
| [Hur gör jag för att lägga till en egen sida med php-kod i wordpress?](#h.u33uvjbmqfyz) | 4 |
| [Hur gör jag för att kontakta databasen i wordpress?](#h.78wow5ir04v9) | 4 |
| [Hur gör jag för att ändra css i wordpress?](#h.y4zlsgsu6ae8) | 4 |
| [Hur gör jag för att installera ett plugin i wordpress?](#h.7wtru0m4beax) | 5 |
| [Hur fungerar wordpress? Var ligger alla filer? Hur sparas data?](#h.qak17sgfqygj) | 5 |
| [Hur går versionshantering till? När gör man det? Hur gör man det?](#h.o5rcwo8nsh0c) | 5 |
| [Diagram för wordpress](#h.293qwouzozpp) | 5 |
| [Hur gör man för att skapa en GET funktion i ett API?](#h.6pokpp8ups6) | 5 |
| [Sökväg till GET funktionsfil ?](#h.odjidcxnpbea) | 6 |
| [Sökväg till POST funktionsfil ?](#h.wl8zhe39x2js) | 6 |
| [Hur gör man för att skapa en POST funktion i ett API?](#h.xgoy7x3gl6cp) | 6 |
| [Fördelar med api](#h.kvscihajldnt) | 7 |
| [Nackdelar med api](#h.fc92y5yya6n3) | 7 |
| [Kodstruktur api](#h.cjb2xoe97cmq) | 8 |

## 

## Arkitekturellt signifikant krav



Spelet postar data om ett achievement har blivit avklarat mot apiet. Apiet kontaktar databasen och returnerar en json sträng som det lyckades eller inte. Spelet kan också hämta data från apiet genom att kontakta det som i sin tur kontaktar databasen. Apiet returnerar då en json sträng med data som innehåller datan man frågade efter. I wordpress så gör vi endast en select direkt från databasen för att kunna visa achievements och dess statusar. Om ett achievement är avklarat så ska det finnas en möjlighet att dela via facebook. Wordpress kontaktar då facebook.

# 

## Hur får jag igång wordpress på min lokala utvecklingsmiljö

Vi använder oss av EasyPHP Devserver när vi jobbar lokalt med wordpress.

1. Hämta hem senaste versionen av EasyPHP
2. Hämta senaste versionen av projektet från github
3. Skapa ett sk. Alias i EasyPHP till sökvägen för projektet (Se till att mappen som wordpressmappen ligger i heter Projektarbeteigrupp)
4. Nästa steg är att i EasyPHP menyn lägga till våran databas
5. Klicka nu på aliaset du skapade. Wordpress borde då köras.

Nu borde du kunna börja koda.

## Vilka api-er som används

Vi använder vårat eget api.

## Vilka plugins som används i wordpress

* Simplr User Registration Form Plus (Formulär för registrering)
* bbPress (Plugin för forum)
* Akismet (Plugin som skyddar mot spam)
* Very Simple Contact Form (Plugin för kontaktformulär)
* Easy Updates Manager (Plugin som man kan hindra wordpress från att uppdatera sig själv eftersom det inte är så bra om en uppdatering kommer efter redovisningsdagen som gör att något plugin inte fungerar som det ska)

## Kodstruktur wordpress

Wordpress består i stort sett av sekventiell php kod vilket vi också använder oss av för att hålla oss till samma struktur och uppbyggnad som wordpress.

## Fördelar med wordpress

Vi får ett färdigt projekt som vi kan bygga på med fler funktioner och anpassa som vi vill ha det. Vi kan nu lägga mer fokus på att göra en koppling mellan spelet och vårt api. Wordpress innehåller saker som spamhantering, säkerhetsfunktioner och en massa andra smidiga funktioner som annars skulle ta ganska lång tid att implementera och göra det bra. Det finns väldigt många plugins till wordpress vilket gör att det blir ganska enkelt att lägga till vissa funktioner som inte finns från början.

## Nackdelar med wordpress

Wordpress är ett ganska så stort projekt med c:a 1400 filer. Det tar ett tag att komma in i var de olika funktionerna, filerna osv. finns. Wordpress är även ett projekt med öppen källkod vilket gör att säkerhetsbrister som någon hittat kan användas för alla wordpress-sidor.

## Bra klasser, filer och funktioner att känna till innan du börjar

Den globala variablen $wpdb kan vara mycket bra att känna till då denna används för att ansluta till databasen. Det finns även en global som heter $current\_user som används för att hämta information om den inloggade användaren. Det finns en php-fil som heter functions.php som innehåller en stor mängd funktioner som ofta används.

## Något som jag ska tänka på innan jag börjar?

Innan du börjar koda kan det vara bra att kolla igenom dokumentationen för wordpress på deras hemsida

<http://codex.wordpress.org/Main_Page> för att få en överblick av hur olika saker görs och fungerar.

## Hur gör jag för att lägga till en egen sida med php-kod i wordpress?

Vi har gjort på följande sätt. Om du går till sökvägen: Projektarbete-i-grupp-Community\wordpress\wp-content\themes\twentythirteen så finns det där i en php-fil som heter **template\_emptypage.php**. Den här filen fungerar ungefär som en mall för en sida. Högst upp i **template\_emptypage.php** så finner du något i stil med:

/\*

Template Name: EmptyPage

\*/

Här kan du ange vad mallen ska heta. Om du sedan går in i admin-menyn för wordpress och väljer pages->new page så finns det en dropdown där du kan välja mallar för sidan. Din egen sida med php-kod borde nu gå att visas.

## Hur gör jag för att kontakta databasen i wordpress?

Detta är mycket enkelt att göra. Wordpress har en global variabel som du kan ropa på som heter $wpdb.

Sedan är det bara att skriva något i stil med:

$results = $wpdb->get\_results( "SELECT \* FROM wp\_achievements WHERE username = Admin");

Denna funktion finns att kolla lite närmare på [här](https://codex.wordpress.org/Class_Reference/wpdb).

## Hur gör jag för att ändra css i wordpress?

Det finns ett antal css-filer i wordpress. Här är en lista över de viktigaste css-filerna och var de finns:

Om du vill ändra css i forumet så är det denna filen som ska ändras i.

Projektarbete-i-grupp-Community\wordpress\wp-content\plugins\bbpress\templates\default\css\bbpress.css

Om du vill ändra css för övriga wordpress sidor så ändrar du i:

Projektarbete-i-grupp-Community\wordpress\wp-content\themes\twentythirteen\style.css

Om du vill ändra css för registreringsformuläret så ändrar du i:

Projektarbete-i-grupp-Community\wordpress\wp-content\plugins\simplr-registration-form\simplr\_reg.css

För varje plugin du installerar så finns det antagligen en egen css-fil du får leta reda på i mappen för plugin.

## Hur gör jag för att installera ett plugin i wordpress?

Det första sättet att göra detta är att leta upp ett plugin på wordpress hemsida, github eller något annat ställe på internet. Ladda ned .zip filen och lägg sedan den filen i: Projektarbete-i-grupp-Community\wordpress\wp-content\plugins. Nästa steg är att gå in i admin menyn i wordpress och välja plugins->installed plugins. Det plugin du nyss lade till i mappen ovan borde nu finnas listat här. För att aktivera det tryck på activate och för att avaktivera tryck på deactivate. Nu borde du kunna använda dig av pluginet. Om det inte skulle fungera så kan det vara en bra idé att extrahera filerna i zip-filen.

Ett annat sätt att göra detta på är att i admin menyn gå in på plugins->add new. Detta tar dig till en sida där du kan söka och leta efter plugins. Klicka på ett plugin du vill ha och sedan installera och aktivera så är det redo att köras. Här kan du även ladda upp zip-filer och mappar med plugins.

## Hur fungerar wordpress? Var ligger alla filer? Hur sparas data?

Data verkar sparas i databasen. Datan som sparas här är alla poster, inställningar, användare, stats, achievements, länkar osv.

## Hur går versionshantering till? När gör man det? Hur gör man det?

Versionshantering borde ske minst en gång per dag och det görs via github och google docs. Innan du börjar koda så se till att du inte jobbar i samma fil som någon annan håller på med. Du ska alltid synka ner den senaste versionen innan du börjar jobba och när du pushar så se till att din kod inte stör någon annans om den inte fungerar. När du pushar mot github så är det bra om du skriver lite kort om vad det är du har gjort eller om det är något problem.

## Diagram för wordpress

Vi kan tyvärr inte hitta ett fullständigt klassdiagram eller interaktionsdiagram för wordpress. Vi har heller inte tid att skriva sånt med tanke på att det är c:a 1400 filer + alla filer för plugins.

## Hur fungerar användarrollerna i wordpress? Vad för rättigheter?

<https://codex.wordpress.org/Roles_and_Capabilities>

## Hur gör man för att skapa en GET funktion i ett API?

Börjar med att implementera några POST variabler som tar emot användarnamn, uppdrag, och status på vad man önskar att få reda på.

Exempel:$username = isset($\_GET['username']) ? $\_GET['username'] : "";

Gör en if sats som kollar om achievement och användarnamn inte är tomma o kolla om statusen är tom isådana fall kontakta databasen och ta reda på om det här uppdraget finns eller inte på den här användaren.

Exempel: if(!empty($username) && !empty($achievementName) && empty($status))

Om raden finns returnera ett json meddelande med namn uppdrag och statusen på uppdraget, 1 betyder att uppdraget är klart och 0 betyder att det inte är avklarat. Annars skicka felmeddelande att uppdraget ej existerar på denna användaren. Det här är för att hämta ut en specifik status för ett uppdrag på en viss användare.

När man ska hämta alla användare så ska en if sats implementeras med följande villkor kolla så att användarnamnet inte är tomt sedan ska status vara tomt och likaså uppdraget. I sådana fall kontakta databasen och välj alla uppdrag på denna användaren gör en foreach sats och loopa ut användarnamn, status på uppdraget och uppdragen som finns på den användaren. Annars kasta felmeddelande att det inte finns några uppdrag på denna användaren.

## Sökväg till GET funktionsfil ?

Lokalt följande adress: 127.0.0.1/achievements/info.php

första är lokalhostens adress sedan är achievements ett alias som kopplar till denna fysiska adressen: C:\Users\<användare>\Documents\GitHub\Projektarbete-i-grupp-Community\api\achievements och har namnet achievements.

Externt följande adress: www.dindoman.com/api/achievements/info.php

Vilka parametrar behöver jag ändra för att ansluta mot databasen i API:et?

Gå in i <path>/api/achievements/confi.php

Ändra host, databasnamn, användarnamn och lösenord till dina egna uppgifter för din databas.

## Sökväg till POST funktionsfil ?

Lokalt följande adress: 127.0.0.1/achievements/status.php

första är lokalahostens adress sedan är achievements ett alias som kopplar till denna fysiska adressen i vårt fall: C:\Users\<användare>\Documents\GitHub\Projektarbete-i-grupp-Community\api\achievements och har namnet achievements.

Externt följande adress: www.dindoman.com/api/achievements/status.php

Vilka parametrar behöver jag ändra för att ansluta mot databasen i API:et?

Gå in i <path>/api/achievements/confi.php

Ändra host, databasnamn, användarnamn och lösenord till dina egna uppgifter för din databas.

## Hur gör man för att skapa en POST funktion i ett API?

Börja med att deklarera följande variabler användarnamn, lösenord, uppdrag och om uppdraget är klart.  
Exempel: $achievement = isset($\_POST['achievement']) ? $\_POST['achievement'] : "";

Gör sedan en if sats som kollar om det har skickats något annars skicka ett felmeddelande.

Exempel: if($\_SERVER['REQUEST\_METHOD'] == "POST")

Eftersom lösenordet ska skickas krypterat så finns det en funktion som decryptar denna till ett vanligt lösenord som sedan krypteras för att matcha wordpress kryptering.

Exempel: $decrypted = base64\_decode($password);

$password1 = mcrypt\_decrypt(MCRYPT\_RIJNDAEL\_256, $key, $decrypted, MCRYPT\_MODE\_CBC, $iv);

$pass = preg\_replace(

array(

'/\x00/', '/\x01/', '/\x02/', '/\x03/', '/\x04/',

'/\x05/', '/\x06/', '/\x07/', '/\x08/', '/\x09/', '/\x0A/',

'/\x0B/','/\x0C/','/\x0D/', '/\x0E/', '/\x0F/', '/\x10/', '/\x11/',

'/\x12/','/\x13/','/\x14/','/\x15/', '/\x16/', '/\x17/', '/\x18/',

'/\x19/','/\x1A/','/\x1B/','/\x1C/','/\x1D/', '/\x1E/', '/\x1F/'

),

array(

"", "", "", "", "",

"", "", "", "", "", "",

"", "", "", "", "", "", "",

"", "", "", "", "", "", "",

"", "", "", "", "", "", ""

),

$password1

);

Viktigt att man efter detta krypterar lösenordet som man tagit emot till wordpress kryptering för att detta ska stämma mot databasen.

Exempel: $wp\_hasher = new PasswordHash(8, TRUE);

$check = $wp\_hasher->CheckPassword($pass, $hash);

Viktigt att man importerar in include\_once('PasswordHash.php'); sedan är variabeln $pass det lösenordet man dekrypterade innan (se ovan).

Där $iv och $key är nycklar för att låsa upp krypteringen och denna skickas från det stället som vill spara informationen i databasen, i vårt fall från unity där de utvecklar spelet. Sedan kollar man så inte användarnamn, lösenord och uppdrag är tomma om de inte är tomma så kollar man mot databasen om användaren existerar i användartaballen om denna gör det. Kolla då mot uppdragstaballen om användaren och uppdraget existerar och om det gör det så uppdatera statusen på uppdraget till det man tagit emot. Annars lägg till uppdraget på användaren med den status på uppdraget som man tagit emot. Om användaren inte existerar i användartabellen så skicka felmeddelande.

## Fördelar med api

* Kodoberoende. Spelet kör c# kod medans vårt api kör PHP och likaså wordpress. De kan kontakta varandra.
* Spelet slipper kontakta databasen direkt vilket är effektivare

## Nackdelar med api

* Lite långsammare prestanda
* Säkerheten kan vara sämre

## Kodstruktur api

Vårt api är skrivet med ungefär samma kodstandard som wordpress. Koden är till stor del sekventiell kod.