

## Inlämningsuppgift 2: Dog Breed Quiz API

Detta är en individuell inlämningsuppgift som efter inlämning även ska redovisas muntligt. Uppgiften grundar sig i att ni får en färdig sk. *frontend* (dvs. HTML/CSS/JS) att tillgå, där ni ska undersöka kodbasen för att lista ut hur en viss *backend* (ett JSON-API) ska byggas. Ni kommer även att få åtkomst till en komplett (online) version av denna webbapplikation, så att ni både kan se slutresultatet (dvs. en fungerande *frontend*) men även få möjlighet till att se och testa vad detta JSON-API skickar ut och tar emot.

Ni ska, genom att undersöka både kodbasen och testa online-versionen, bygga ett API i PHP som arbetar med JSON-data. Eftersom ni har en färdig *frontend* att tillgå kan ni även se denna som ett sätt att testa och se om ert API faktiskt fungerar som det ska (och även jämföra mot online-versionen). Hur ni bygger ert API är upp till er, så länge *frontend'en* fungerar som online-versionen.

Avslutningsvis ska ni även skapa dokumentation för detta API, antingen i form av en PDF eller som en simpel webbsida (HTML/CSS). Hur ni utformar detta är även upp till er, men det ska enkelt gå att avläsa vilka resurser det finns, vilken data (JSON) dom antingen tar emot eller skickar ut och vilka fel som hanteras (se nedan).

### Kravspecifikation

Det finns vissa krav vad gäller ert JSON-API som måste uppfyllas formellt sätt.

- Det får endast skrivas i PHP (med korrekt formaterad och indenterad kod)
- Ni måste använda er av relevanta namn på både variabler och funktioner
- Datan (JSON) som skickas och tas emot ska vara densamma som i online-versionen (dvs. ni får **inte** redigera den *frontend* ni har blivit tilldelad)
- Respektive resurs (t.ex. `/api/login.php`) ska minst hantera två fel (eller "felaktiga situationer"), där ett fel kan vara att kontrollera att det är rätt HTTP-metod, att ert API faktiskt tagit emot rätt data eller att en JSON-fil måste finnas
- JSON-data som sparas i fil(er) ska formateras med `JSON_PRETTY_PRINT`

### Material

Ni kommer bli tilldelad en HTML-fil (som innehåller sk. *inline* CSS och JS) och en mapp med hundbilder. **Tänk på** att denna mapp med hundbilder är också grunden till vilka hundar (namnen) vi skapar quiz-frågor om.

### Var ska jag börja?

Det gäller att börja att försöka förstå *frontend* koden, så att ni där kan se vilka resurser som anropas - och samtidigt lista ut hur datan är uppbyggd (som antingen skickas eller tas emot). För att djupare undersöka detta bör ni gå vidare till online-versionen och nyttja nätverksfliken i Firefox's utvecklarverktyg, så ni faktiskt ser vad som skickas och tas emot. Därefter, när ni vet vilka resurser det gäller, kan ni testa att skriva

in dom på egen hand i en webbläsare, kanske till och med testa att skicka något till dom (med bra och/eller dålig data) för att se vad som händer.

Sen gäller det att börja koda sitt API, och tänk på att ni har fria tyglar - det går att lösa uppgiften på många olika sätt. Här kan ni också samtidigt börja fundera över vad som kan behöva sparas i filer (tips: det går ju att logga in på sidan, så förmodligen behöver användare sparas någonstans).

Vad gäller dokumentationen är ett tips att söka upp information om några större tjänster (t.ex. Spotify, Twitter, Facebook) och se hur dom dokumenterat sina API, så kan ni enkelt utgå från någonting (dokumentationens utformning behöver inte vara originellt).

## Redovisning

Under redovisningen kan vi komma att ställa frågor kring hur *frontend*-koden fungerar (och därmed hur ni lyckats komma fram till saker och ting) samt hur er *backend*-kod fungerar (ert API) och hur ni tänkt kring er lösning. Frågor kan vara i ren allmänhet eller specifika om en eller flera rader kod.

Ett tidsschema kommer att publiceras under samma vecka som redovisningen.

## Inlämning

När ni är färdiga och ert API fungerar som det ska så paketerar ni det (PHP-filerna) tillsammans med er dokumentation i form av en zipfil, och lämnar sedan in det på Canvas.

**Lycka till!**