

36) Vervang de init functie in de AppDelegate.h door de volgende code:

```
- (id)init {
    self = [super init];
    if(self != nil) {
        self.deviceToken = @"";
        self.payload = @"{"aps":{"alert":{"News!},"badge":5,"sound":{"Blip.wav"},"acme1":{"bar"},"acme2":{"42}}";
        self.certificate = [[NSBundle mainBundle] pathForResource:@"aps_developer_identity" ofType:@"cer"];
    }
    return self;
}
```

37) Als dit werkt kun je door naar de volgende stap om te zorgen dat een serversided script het Push bericht gaat sturen naar de Apple Notification Server.

Server sided

38) Open de Keychain op je mac en exporteer de Apple Development Push Services certificate. Bewaar 'm als apns-dev-cert.p12

39) Exporteer nu ook de Private Key waar de Apple Development Push Services certificate onder valt. Bewaar deze als apns-dev-key.p12

40) Deze twee keys moeten nu geconverteerd worden naar het PEM formaat. Voer in de terminal de volgende twee commando's uit op de files. Voer het Import password in en gebruik als PEM phrase ook (eventueel?) dat wachtwoord.

```
openssl pkcs12 -clcerts -nokeys -out apns-dev-cert.pem -in apns-dev-cert.p12
```

```
openssl pkcs12 -nocerts -out apns-dev-key.pem -in apns-dev-key.p12
```

41) Plak de twee PEM bestanden aan elkaar mbv het cat commando:

```
cat apns-dev-cert.pem apns-dev-key.pem > apns-dev.pem
```

42) Maak nu een test PHP scriptje aan voor de payload die naar de Apple Notification Server gestuurd moet worden:

```
<?php
$payload['aps'] = array('alert' => 'This is the alert text', 'badge' => 1, 'sound' => 'default');
$payload['extra'] = array('data' => 'Je mag extra dictionaries toevoegen met een willekeurige naam.',
                        'warning' => 'max 256 bytes alles in totaal', 'id' => 5);
$payload = json_encode($payload);
echo $payload;
?>
```

De output hiervan is een JSON string:

```
{ "aps": { "alert": "This is the alert text", "badge": 1, "sound": "default" },
  "extra": { "data": "Je mag extra dictionaries toevoegen met een willekeurige naam.", "warning": "max 256 bytes alles in totaal", "id": 5 }
}
```

De uiteindelijke PHP code is voor de sandbox server. // for production change the server to ssl://gateway.push.apple.com:219

```
<?php
$payload['aps'] = array('alert' => 'This is the alert text', 'badge' => 1, 'sound' => 'default');
$payload = json_encode($payload);

$apnsHost = 'gateway.sandbox.push.apple.com';
$apnsPort = 2195;
$apnsCert = 'apns-dev.pem';
$deviceToken = 'XXXXXXXXXXXXXXXXXXXXXXXXX';
$pass = 'XXXXXXXXX';

$streamContext = stream_context_create();
stream_context_set_option($streamContext, 'ssl', 'local_cert', $apnsCert);
stream_context_set_option($streamContext, 'ssl', 'passphrase', $pass);

$apns = stream_socket_client('ssl://' . $apnsHost . ':' . $apnsPort, $error, $errorString, 2,
STREAM_CLIENT_CONNECT, $streamContext);
$apnsMessage = chr(0) . chr(0) . chr(32) . pack('H*', str_replace(' ', '', $deviceToken)) . chr(0) . chr(
strlen($payload)) . $payload;
fwrite($apns, $apnsMessage);

socket_close($apns);
fclose($apns);

echo "$error $errorString";
?>
```