

meetup



CHANGE THE GAME

Authentication with Laravel Passport & Vue.js

*Philip Peinsold
CTO & Co-Founder*

Authentication with Laravel Passport & Vue.js

- Content

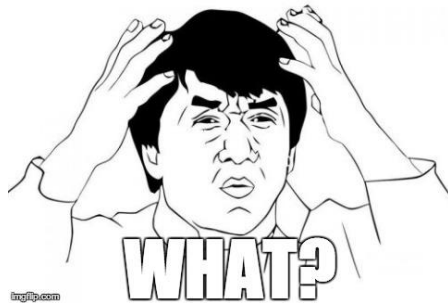
- Backend – Laravel Passport
 - What is it
 - Why using it
 - How to use it
- Frontend – Vue.js
 - Get Access Token
 - Store Access Token
 - Use Access Token



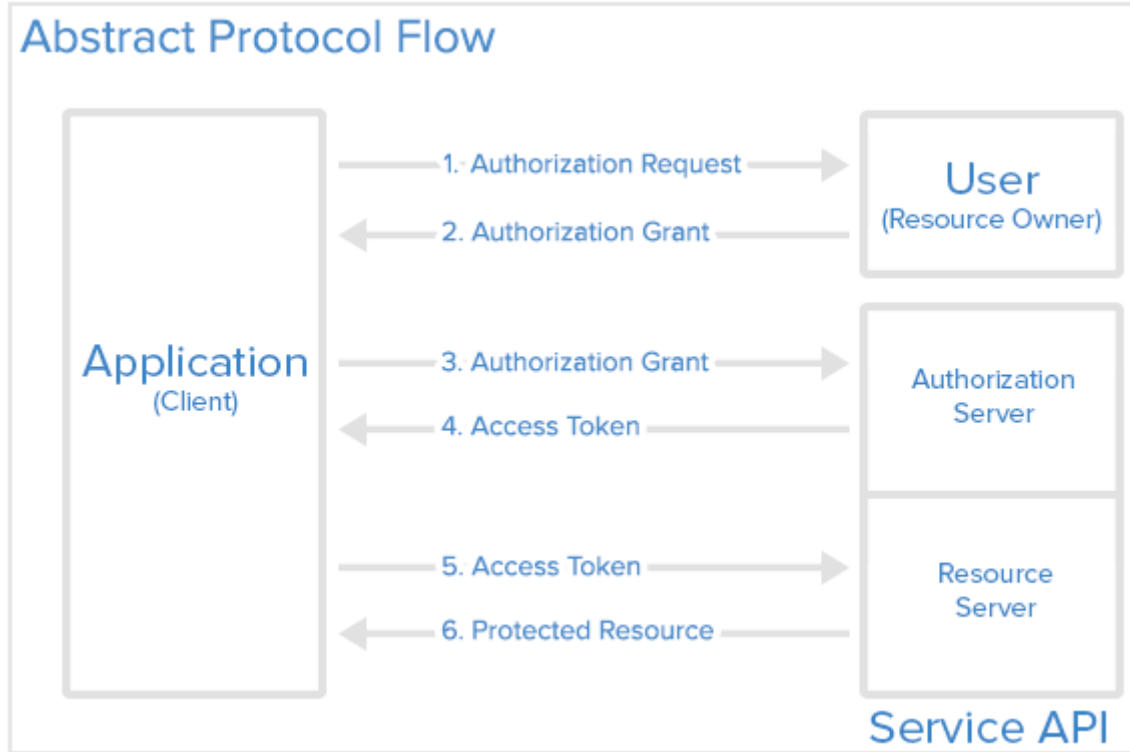
Laravel Passport

- What is Passport?

- OAuth2 is the reigning ruler of authentication
- Implementation is complex & difficult
- Passport = full OAuth2 server implementation
- Uses the „League OAuth2“ Open Source Project
- Provides easy-to-learn & easy-to_implement syntax on top



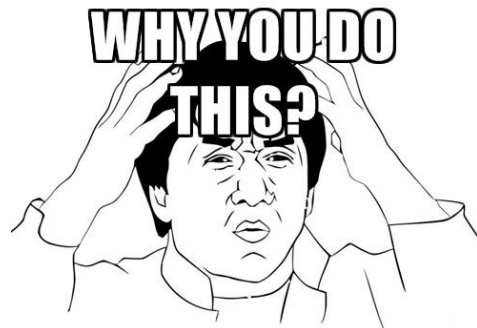
Short explanation of OAuth2



Laravel Passport

- Why/When use Passport?

- Separate backend and frontend
- Need to get data without sessions
- APIs typically use tokens to authenticate users
- Need to secure your API
- Give others a secure way to consume your API



Installation & Setup of Passport

- Install passport via composer
 - `composer require laravel/passport`
- Migrate Passport database
 - `php artisan migrate`
- Create encryption keys and clients
 - `php artisan passport:install`
- Extend functionality of your user model
 - Add `Laravel\Passport\HasApiTokens` to `App\User` model
- Register passport routes
 - Call `Passport::routes()` in `boot()`-Method of `AuthServiceProvider`
- Config authentication driver
 - In `config/auth.php` set driver option of the api authentication guard to passport



CORS - Problem

- Cross-origin resource sharing Error

- `access-control-allow-origin"`, "XMLHttpRequest cannot load `http://****:8000/oauth/token`. Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://localhost:8100' is therefore not allowed access.

- Use a middleware for routes

```
Route::group(['middleware' => 'cors'], function() {  
    Passport::routes();  
});
```

- Implement middleware

```
class Cors {  
    public function handle($request, Closure $next) {  
        return $next($request)  
        ->header('Access-Control-Allow-Origin', '*')  
        ->header('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, OPTIONS')  
        ->header('Access-Control-Allow-Headers', 'Origin, Content-Type, Authorization, Language');  
    }  
}
```



Short Demo (Postman)

- Get access token
 - `POST /oauth/token`
- Use access token
 - `GET /user`



Vue.js – Get Access token

- Use the OAuth Password Grant Client
 - Classic user authentication (Email + Password)
- Use Axios to call endpoint: **POST /oauth/token**
- Important data:
 - grant_type
 - client_id
 - client_secret
 - username
 - password



Vue.js - Get Access Token

- `async login(user: User, password: string): Promise<OAuthResponse> {`
 `return axios.post(process.env.API_ENDPOINT + '/oauth/token', {`
 `grant_type: process.env.GRANT_TYPE,`
 `client_id: process.env.CLIENT_ID,`
 `client_secret: process.env.CLIENT_SECRET,`
 `username: user.email,`
 `password: password`
 `})`



Vue.js - Store Token in local storage

```
export const AccessToken = {  
  get: {  
    accessToken(): string | any {  
      return localStorage.getItem(KEY_TOKEN);  
    }  
  },  
  
  set: {  
    accessToken(token: string) {  
      localStorage.setItem(KEY_TOKEN, token);  
    }  
  }  
};
```



Vue.js – Login and save token

```
login(user: User, password: string) {  
  return axios.post(process.env.API_ENDPOINT + '/oauth/token', {  
    grant_type: process.env.GRANT_TYPE,  
    client_id: process.env.CLIENT_ID,  
    client_secret: process.env.CLIENT_SECRET,  
    username: user.email,  
    password: password  
  })  
  .then(response => {  
    AccessToken.set.tokenType(response.data.token_type);  
    AccessToken.set.accessToken(response.data.access_token);  
    // response.data.refresh_token - token used for refresh call  
    // response.data.expires_in - lifetime of accesstoken  
    // response.data.token_type - usually 'Bearer'  
  })  
}
```




Vue.js - Use Token for further requests

```
axios.interceptors.request.use(function (config) {  
    config.headers.Authorization =  
    AccessToken.get.tokenType() + ' ' +  
    AccessToken.get.accessToken();  
    return config;  
});
```



Short Login Demo




ACCOUNT LOG IN

Email Address

Password

LOG IN

OR

 Continue with Facebook

[Dont have an account?](#)
[Forgot Password?](#)



Social Login

- What about offering a Facebook-Login?
- Laravel Socialite
 - Package to authenticate with OAuth-providers
 - Supports Facebook, Twitter, LinkedIn, Google, GitHub and Bitbucket
- But how do we get an access token?
 - Socialite only provides data & token from facebook API



Login with Socialite

- Place credentials for provider into config/services.php

```
'github' => [  
    'client_id' => env('GITHUB_CLIENT_ID'),      // Your GitHub Client ID  
    'client_secret' => env('GITHUB_CLIENT_SECRET'), // Your GitHub Client Secret  
    'redirect' => 'http://your-callback-url',  
]
```

- Routing

- Redirecting the user to the OAuth provider
- `Route::get('login/facebook', 'Auth\SocialLoginController@redirectToProvider');`
- Receiving the callback from the provider after authentication
- `Route::get('login/facebook/callback', 'Auth\SocialLoginController@handleProviderCallback');`



Login with Socialite

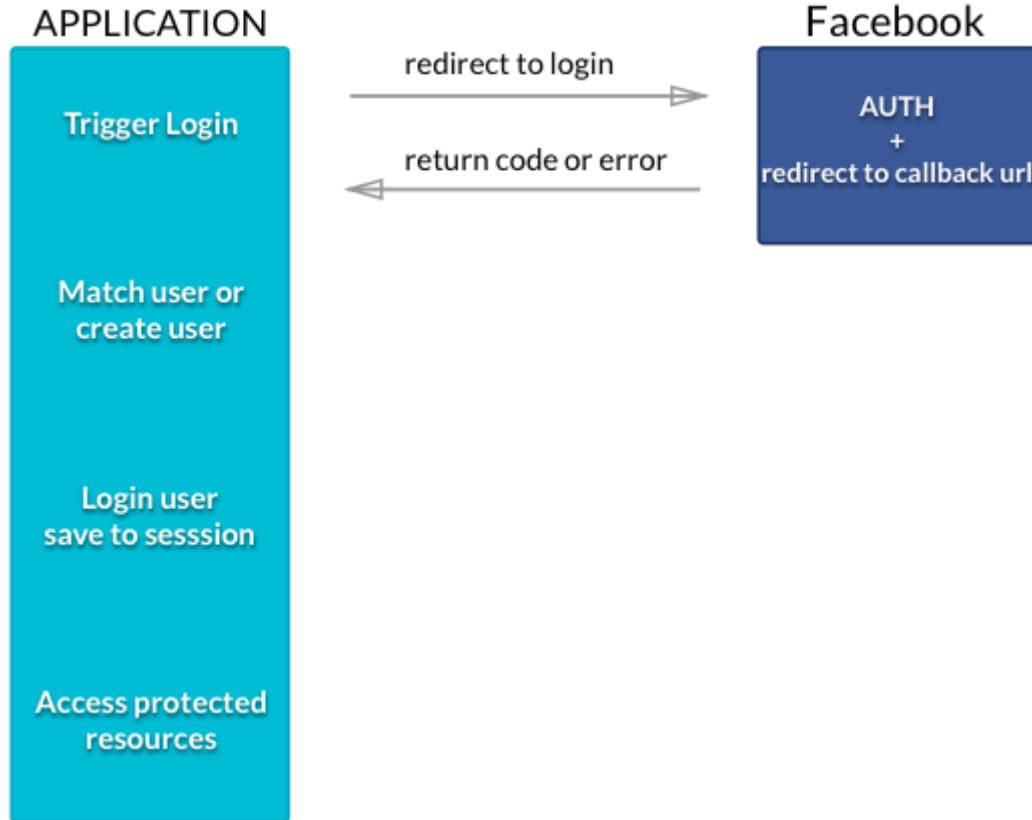
- Create a Controller

```
class SocialLoginController extends Controller
{
    public function redirectToProvider() {
        return Socialite::driver('facebook')->redirect();
    }

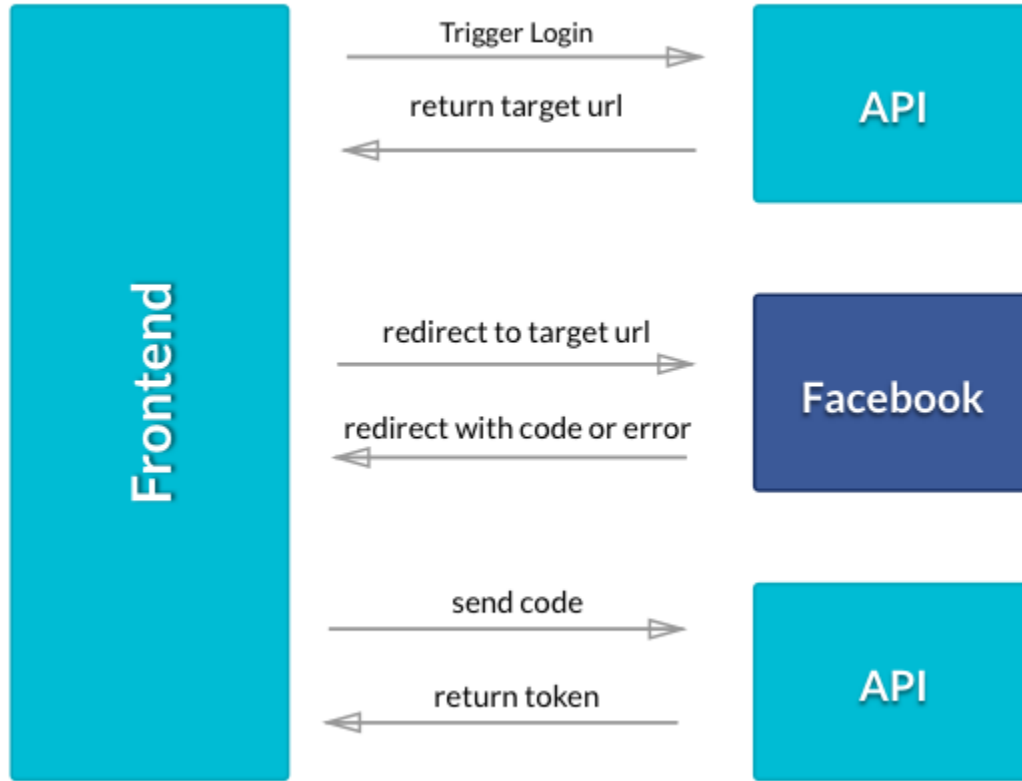
    public function handleProviderCallback() {
        $user = Socialite::driver('facebook')->user();
        // $user->token;
    }
}
```



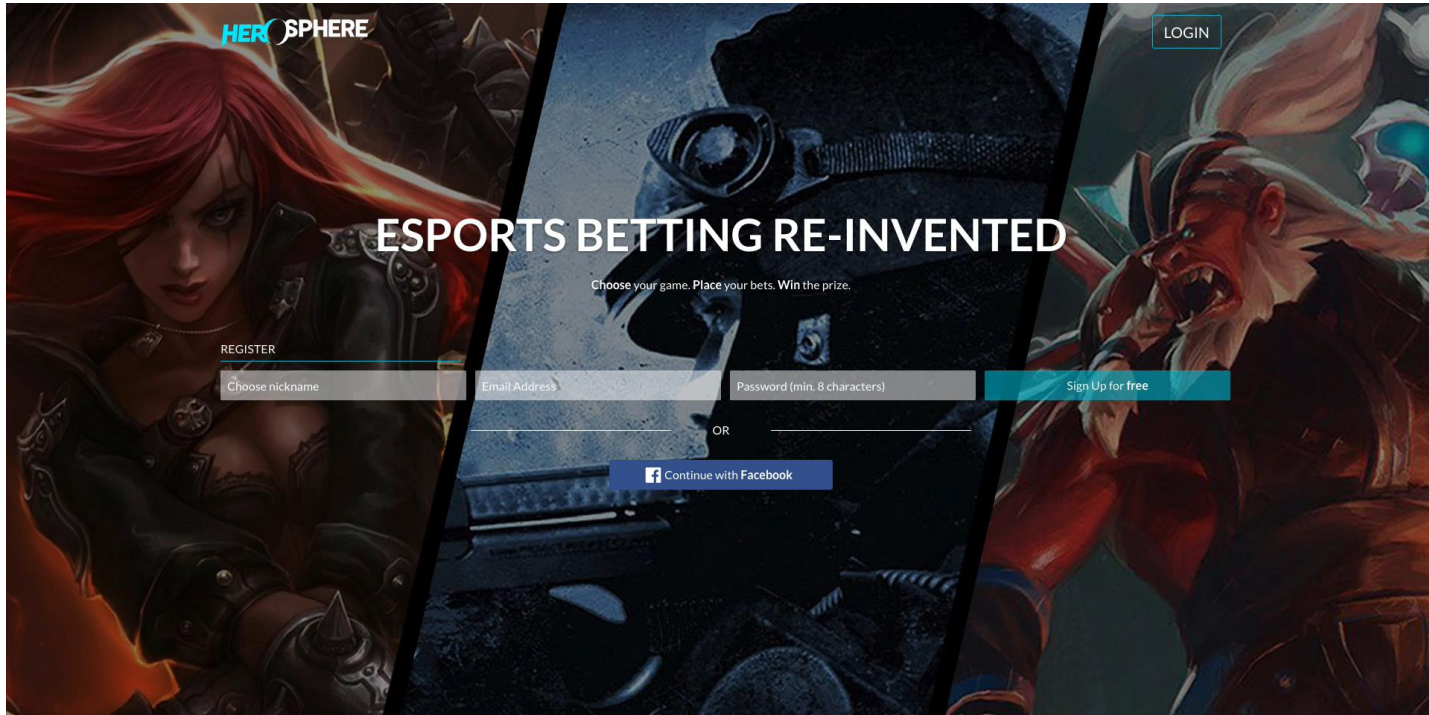
Workflow with Socialite



Workflow Socialite with Passport



Short Facebook Login Demo



meetup



Thank you for your attention!

*Philip Peinsold
CTO & Co-Founder*

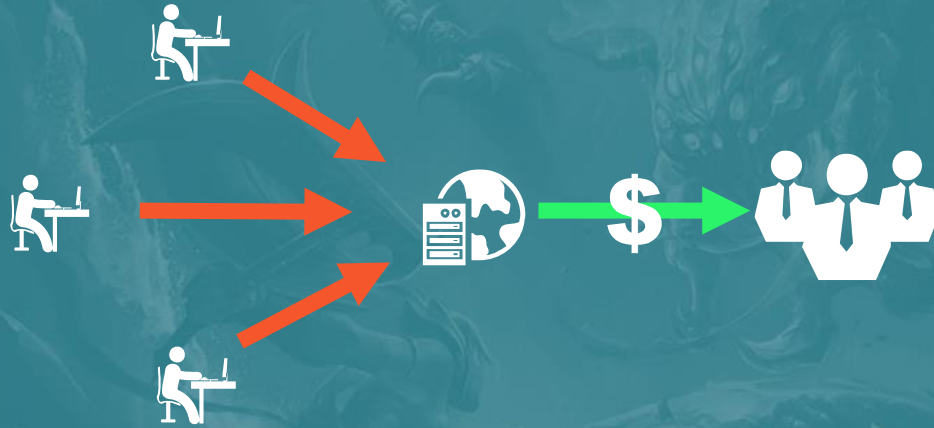
meetup



CHANGE THE GAME

*Philip Peinsold
CTO & Co-Founder*

pain: online sports betting is disappointing



- the inherent social character of a bet is lost
- unfair conditions: users bet against professionals (bookies)

solution: herosphere - challenge your friends



- player vs. player system
- users challenge their friends or the community instead of bookies
- a game changer

eSports



*market is anticipated to grow by a factor of 20
untapped by betting providers*

eSports
cash gambling

\$0.6bn
2016

\$11bn
2020

>200% CAGR

eSports
enthusiasts

90m
2014



180m
2019



>11% CAGR

eSports
becomes mainstream



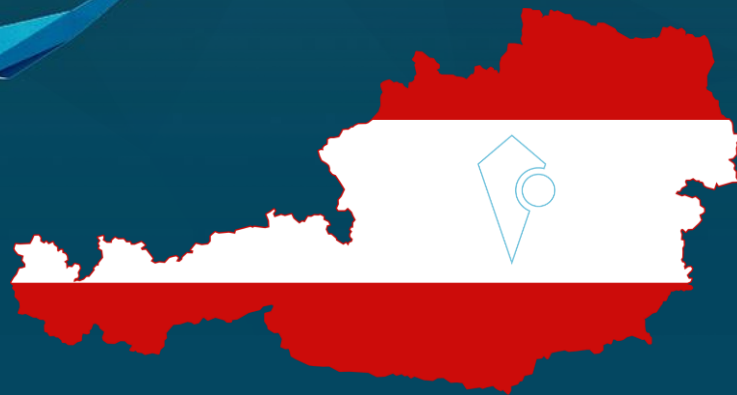
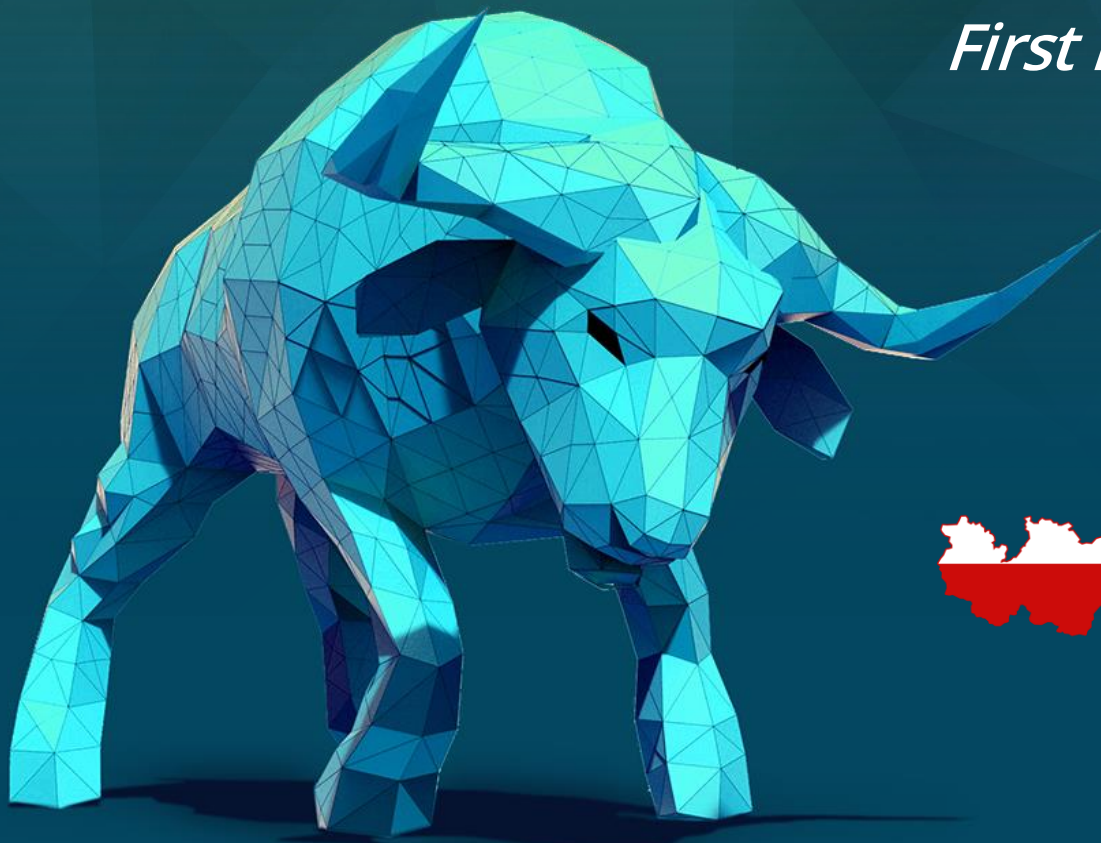
HEROSPHERE

trust through blockchain

- *transparency*
- *security*
- *trust*
- *open-source*



First ICO of Austria



HERO ICO

- *first ICO in Austria*
- *more than 2,000 transaction*
- *~ 2-3 ETH per transaction*
- *USD 2m*



based on a product with more than 250k users
herosphere – your daily fantasy prediction



*> 300.000
registered users*



*> 1.600.000
placed bets*



*herosphere.gg will be the first provider to implement HERO.
More providers will follow.*



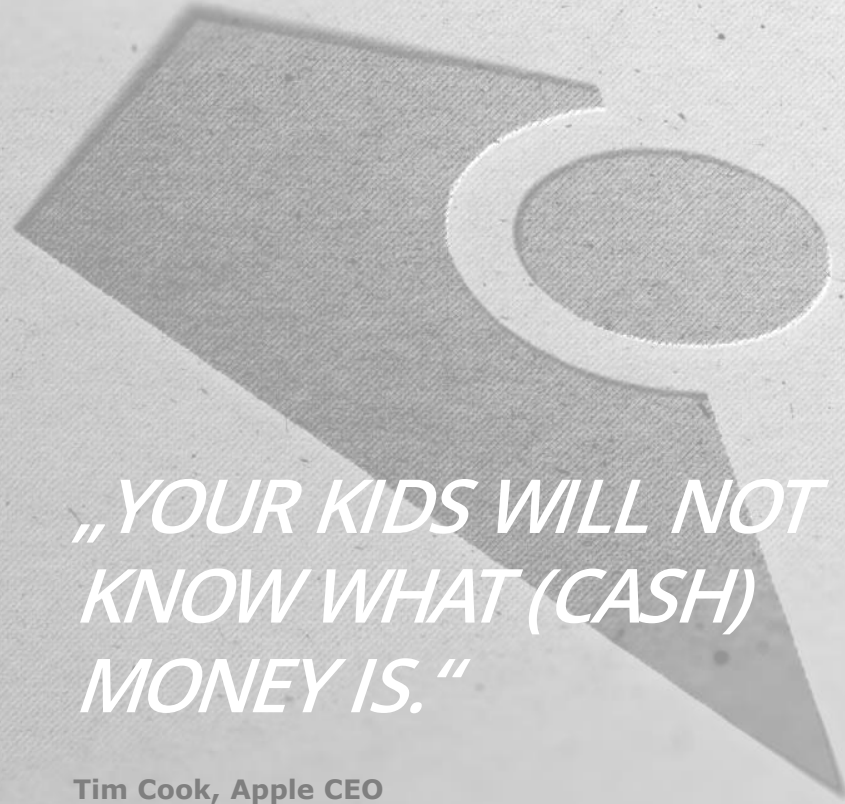
HERO SPHERE

We currently hunting for new HEROs!

- *About you*
- *Experienced in developing professional software*
- *Great PHP/Laravel and/or Vue.js knowledge*
- *Interest into crypto and eSports*
- *Team player*

- *About us*
- *Young and motivated team*
- *StartUp atmosphere with a positive working climate*
- *Employee goodies (free coffee, team trips and team building activities)*





*„YOUR KIDS WILL NOT
KNOW WHAT (CASH)
MONEY IS.“*

Tim Cook, Apple CEO