

Reinventing Engaging Networks actions with ENS REST API

Iain Ure - Frontend Developer

30th JUNE 2020

Today we will cover

1. Engaging Networks pages
2. Other options
3. Example using ENS REST API
4. What is ENS REST and how does it work?
5. Process flow
6. Authentication server
7. Browser application code
8. Questions

Engaging Networks Pages

- Look like [this](#)
- Our HTML and CSS
- HTML structure
 - Header, <form>, Footer
 - Built with dashboard users in mind
 - Maintain consistency across multiple page types
- Every user input maps to supporter database column or question
- Submit button at the end
- JS for
 - Validation
 - Dependencies
 - Captcha
 - Others dependent on page type
- Hosted by Engaging Networks
 - Subdomain
- HTML template for branding or simulate your website

“Roll your own”?

When do our hosted pages become restrictive?

- Use your own HTML, JS and CSS
 - Complex layouts
 - Complex dependencies between fields
- Single page action with multiple stages
 - Donation pages
 - No reloads
- Built-into website
- Part of a wider webapp
 - Game, map, dashboard
 - React, Vue, Angular, Svelte
- Outside the browser e.g. native app

What are the options?

Enhance Engaging Networks page

- Build “on-top” of our HTML, CSS, JS
- HTML template, code blocks
- Depends on size of job
- Developer experience
- Working around our JS

HTML form POST to page 2

- E.g. to POST -> <https://e-activist.com/page/57936/donate/2>
- Responds in HTML
 - Parse HTML to find errors
- Session validation

Create transaction via ENS REST!

What is ENS REST API?

- REST-style API
- Interact with Engaging Networks from outside the dashboard
- JSON-based
- Pass transactions to Engaging Networks
 - Petitions
 - Donations
 - Email to Target
 - Incl. contact messaging with replacements
- Lots of other services too:
 - Get and update supporter data
 - Question data
 - Profiles

Video on ENS

engagingnetworks.support/knowledge-base/engaging-networks-services-ens/

REST Services

engagingnetworks.support/knowledge-base/engaging-networks-services-rest-services/

Using ENS REST

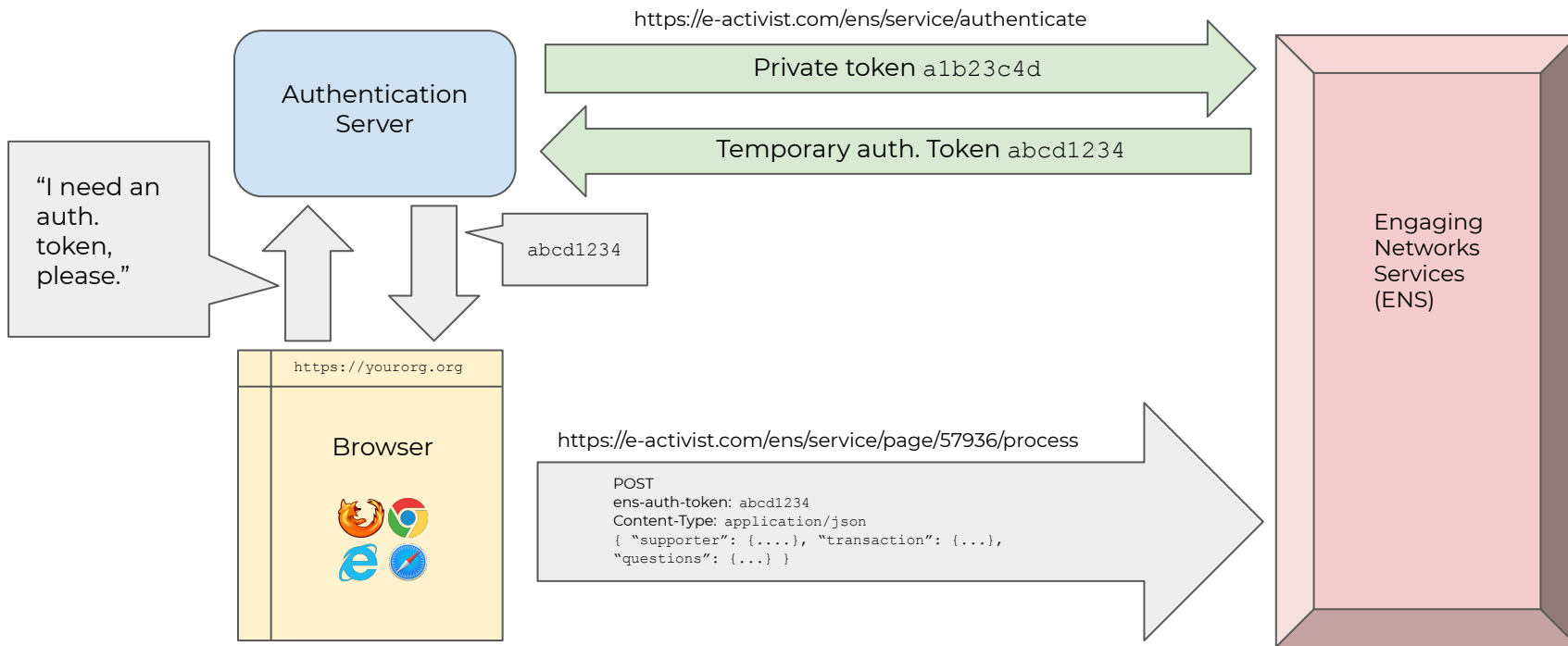
Example

JSON format

JSON passed to Engaging Networks via POST:

```
{
  "supporter": {
    "Email Address": "iain@engagingnetworks.net",
    "First Name": "Iain",
    "Last Name": "Ure",
    "Address 1": "1 Test Street",
    "City": "Testerfield",
    "Region": "Testerton",
    "Postcode": "e123aa",
    "Country": "GB"
  },
  "transaction": {
    "paymenttype": "VISA",
    "donationAmt": "20.00",
    "ccnumber": "4222222222222220",
    "ccexpire": "12/25",
    "recurrpay": "Y",
    "recurrfreq": "MONTHLY",
    "recurrday": 28
  },
  "questions": {
    "Opt In - Newsletters": "Y"
  }
}
```


How does it work?



Authentication server

Why do I need a separate authentication server?

We make two different calls to ENS to pass a transaction to Engaging Networks

Transaction Service


- Data -> ENS
- From any IP address
 - Unlike all other ENS services
 - Cannot obtain supporter data
- Requires a **temporary** auth. Token
 - `ens-auth-token:`
`abcd1234`

Authentication Service

- Private token -> ENS
 - `a1b23c4d`
- Auth. token <- ENS
 - `abcd1234`
 - 1 hour
- Whitelisted IP addresses only
 - Most ENS services are
 - Sensitive supporter data

API User

API iain API User iain+api@engagingnetworks.net

Email Address	<input type="text" value="iain+api@engagingnetworks.net"/>	Status	<input type="text" value="Active"/>
Display Name	<input type="text" value="iain API User"/>	Group	<input type="text" value="encc"/>
First Name	<input type="text" value="iain"/>		
Last Name	<input type="text" value="API User"/>		
Remote Address	<input type="text" value="1.2.3.4~100.200.300.400~991.992.993.994"/> <small>IP Addresses separated by a ~</small>		
API Key	<input type="text" value="489a0e90-6a45-4502-859c-cc10cdcdbd4cd"/>	<input type="button" value="Revoke"/>	<input type="button" value="Copy"/>

Authentication server

Why do I need a separate authentication server?

- Authentication server has a constant IP
- Browser doesn't
 - Unless you whitelist every IP address in the world
- Private token must be kept private
 - If compromised, has to be revoked
 - Breaks all your ENS implementations
- Temporary auth. token
 - Can be seen in AJAX network requests
 - Only lasts an hour
- Good news: auth server it can be re-used for different projects

Authentication server

How do I set one up?

- Secure server you have access to
- Can accept HTTP requests (from browser)
- Can make HTTP requests (to EN)
- Whitelist IP address
- Node JS example
 - An npm package that starts an auth. server running on a particular port
 - Few environmental dependencies - just Node JS
 - Easy to install
 - npm install
- Alternatives
 - PHP cURL
 - Any HTTP server technology
- SSL certificate and key file

Authentication Server

Authentication server example in Node JS

<https://github.com/EngagingNetworks/ens-auth-nodejs-example>

Browser application

- Built in Svelte
 - Like Vue, React, Angular
 - Single file components
 - Output JS is small
- Basic app structure
- API code

Security

- Auth. server example on github
 - Demonstrates concept, starting point
 - HTTPS
 - Not tested to any particular standard
 - Standards and policy for your organisation
- Secure server
- API user must **not** have any “Data management” permissions
 - <https://engagingnetworks.support/knowledge-base/permissions-creating-an-api-user/>
 - <https://engagingnetworks.support/knowledge-base/permissions-creating-permission-groups/>
- Private token must be kept private

Thank You.

Questions and answers

iain@engagingnetworks.net