



hack2020@neilhighley.com



# WITTA

Who exactly knows the person best to advise on tasks and coding problems?

- The person who has committed similar code.
- The person who contributes the most in Confluence, that's who!

Using AI analysis we create a trained model of the code base and ask it.

## Who Is The Task Authority?



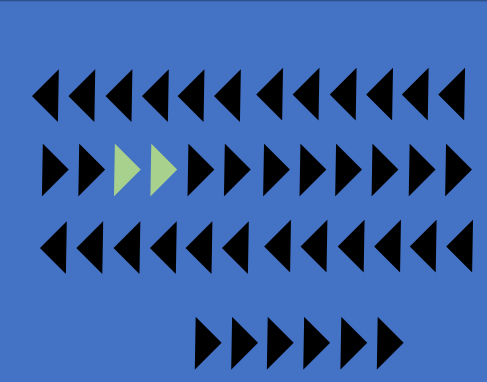
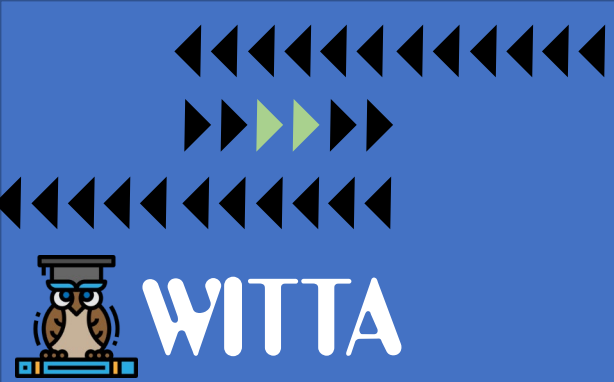
hack2020@neilhighley.com



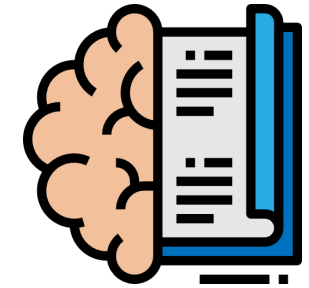
# WITTA

Please note: The AI side of this project is not supplied, as further examination of the dataset required to adequately supply authority models to the AI engine. I have made a suggestion to use AWS but this is not a requirement, any ML platform can be used at the lambda edge. And conversely, any REST API can be used to access it.

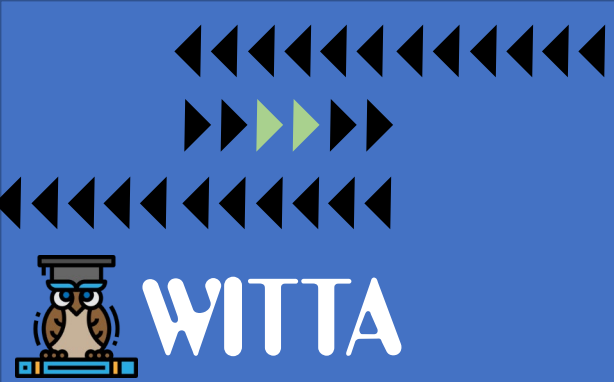
I hope that the explanation within and the proof of concept for the system goes some way to convince of the value of an AI system to supply knowledge during development and how Forge can be utilised to bring together the collected information available in JIRA, Bitbucket and Confluence to better inform teams and encourage collaboration.



hack2020@neilhighley.com

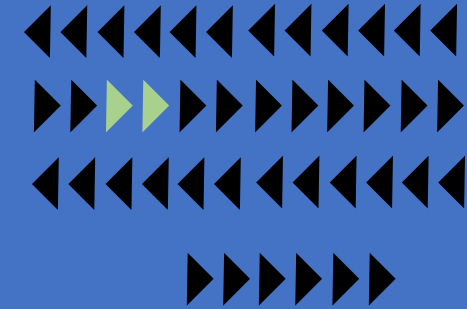
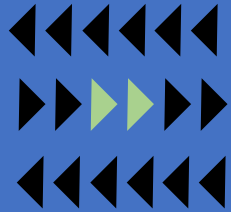


- An AI Authority running on AWS AI models and DataSets, supplying knowledge on task content and supplied queries.
- An AWS API allows the JIRA, Bitbucket and Confluence Applications to populate the model via webhooks on submit and action updates.



ATLASSIAN

codegeist



hack2020@neilhighley.com



## AWS Machine Learning

AWS pre-trained AI Services provide ready-made intelligence for your applications and workflows. AI Services easily integrate with your applications to address common use cases such as personalized recommendations, modernizing your contact centre, improving safety and security, and increasing customer engagement. Because we use the same deep learning technology that powers Amazon.com and our ML Services, you get quality and accuracy from continuously-learning APIs.

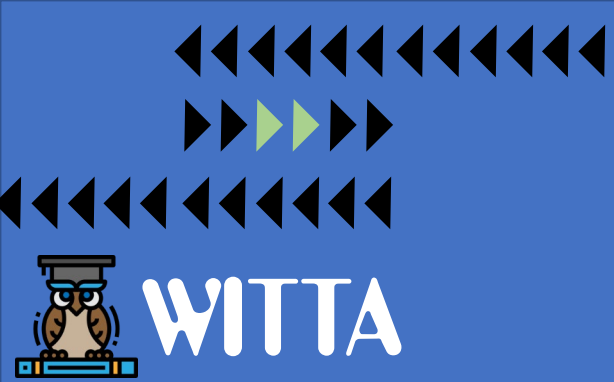
**NOT IMPLEMENTED**

### Amazon Personalise

Amazon Personalize uses machine learning algorithms to create recommendations that respond to the specific needs, preferences, and changing behaviour of your users in real-time. These algorithms also address common complex problems such as creating recommendations for new users or products with no historical data, and popularity biases.

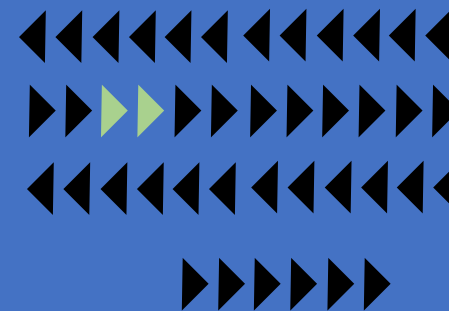
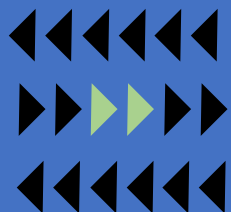
### Amazon Comprehend

Powered by state-of-the-art machine learning models, Amazon Comprehend can discover insights from unstructured text like social media posts, emails, and web pages.



ATLASSIAN

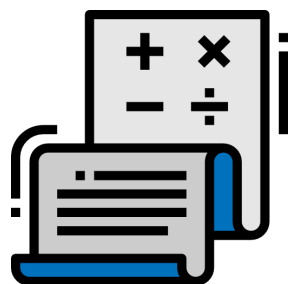
codegeist



hack2020@neilhighley.com



- Every Commit automatically supplies the AI Authority with code comments and class names
- Every Commit automatically supplies the AI Authority with the commit messages
- Every JIRA task will automatically supply the AI authority with content information

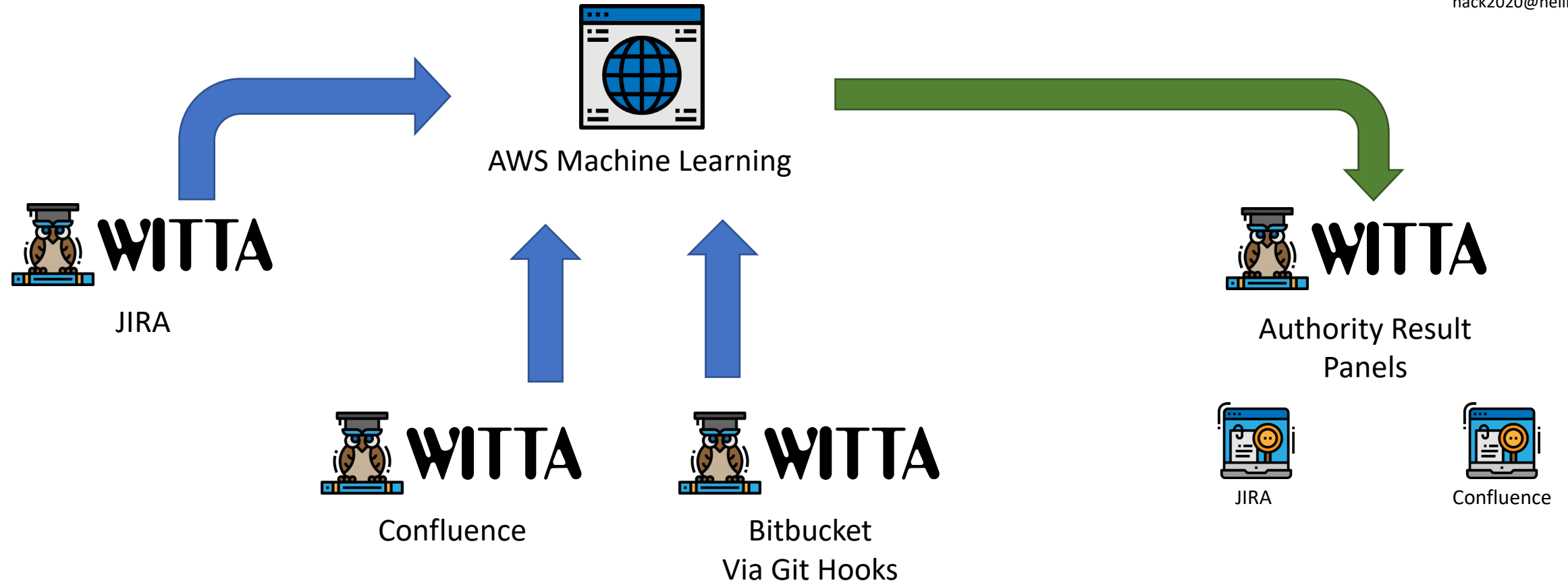


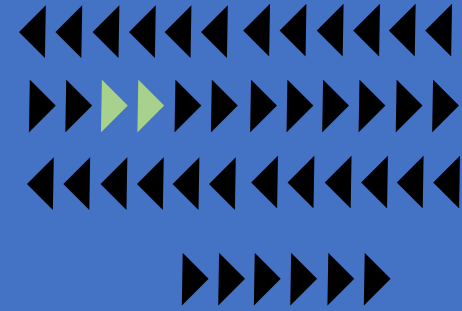
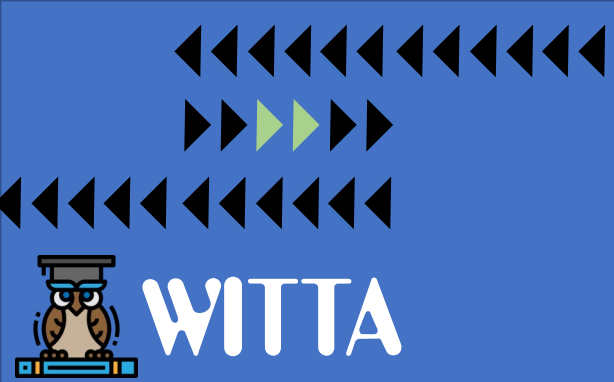
- Confluence activity is automatically supplied to the AI authority
- Confluence user selected text, queries the AI authority
- Confluence can supply candidates for authority with key phrases manually.

# ATLASSIAN codegeist



hack2020@neilhighley.com

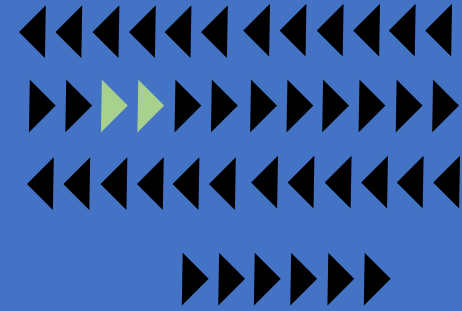
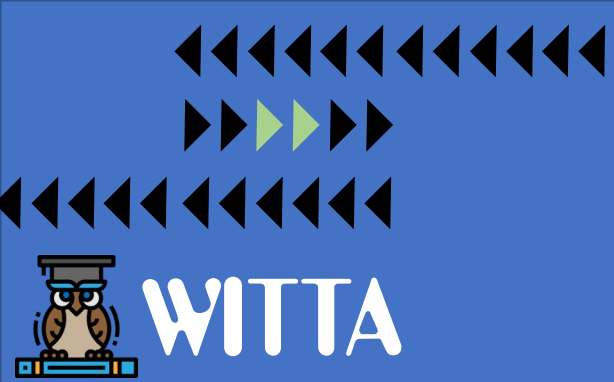




## (Forge UI) Information panels in JIRA

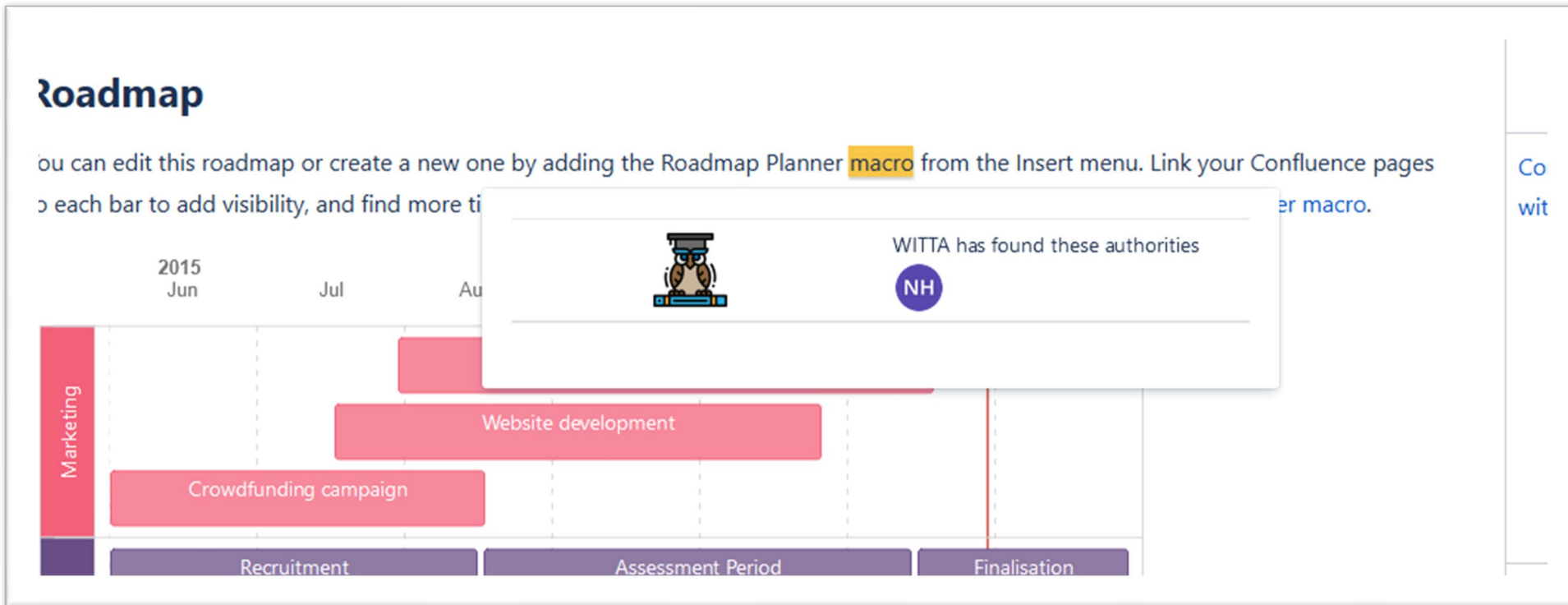
hack2020@neilhighley.com

The screenshot shows a JIRA issue page for a task named "stuff". The issue is in the "To Do" status and is assigned to "Unassigned". The reporter is "neil highley" (NH). The issue was created and updated 47 minutes ago. The description field is empty, with the placeholder text "Add a description...". The "The Authorities in your Team (DEVELOPMENT)" section shows a list of team members, with "NH" (neil highley) being the only one visible. The "Activity" section shows a comment input field with the placeholder text "Add a comment..." and a "Pro tip: press M to comment" message. The page also includes a "To Do" dropdown menu, an "Assignee" field, a "Labels" field (set to "None"), and a "Reporter" field.



## (Forge UI) WITTA query panel in Confluence

hack2020@neilhighley.com







hack2020@neilhighley.com

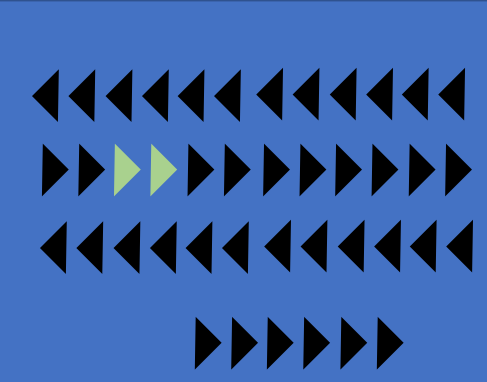
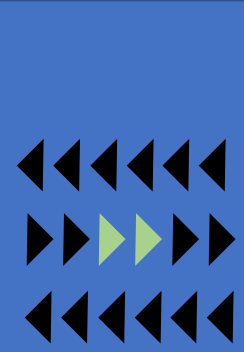
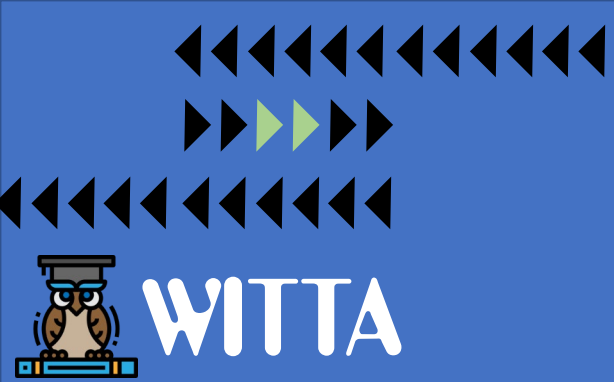


# WITTA

- Automatic AI powered Task Authority Suggestions
- Powerful Edge based intelligence gathering
- Flexible scaling and stack maintenance with a Platform Agnostic API Layer
- Wide ranging future scope; Portable Apps, Team Creation Tools, Mentor Scoring

[HTTPS://WITTA.ATLASSIAN.NET](https://witta.atlassian.net)

[HTTPS://hack2020.appmecha.co.uk](https://hack2020.appmecha.co.uk)



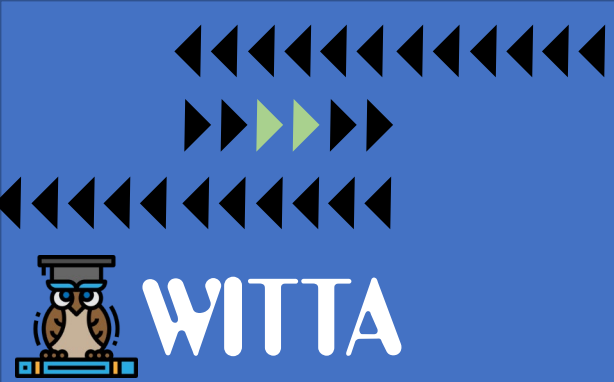
# (Forge UI) WITTA query panel in Confluence

hack2020@neilhighley.com

```
import ForgeUI, {
  Text, InlineDialog, Table, Cell, Row, Image, Avatar, AvatarStack
} from "@forge/ui";

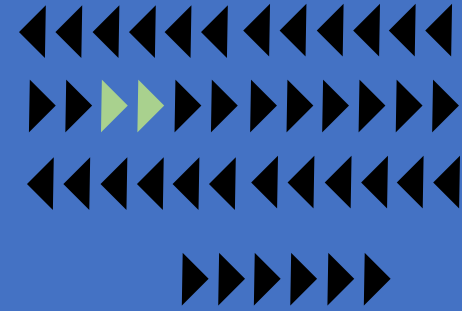
export const FakeWittaAuthorities={()=>{
  return(
    <InlineDialog>
      <Table><Row><Cell><Image alt="witta logo" src="http://hack2020.appmecha.co.uk/002-wisdom-sml.png"/></Cell>
      <Cell><Text>WITTA has found these authorities</Text>
      <AvatarStack>
        <Avatar accountId="5efe82df3404690bae97dd5c"/>
      </AvatarStack>
    </Cell></Row></Table>
    </InlineDialog>
  );
};

export const FakeWittaNoAuthorities={()=>{
  return(
    <InlineDialog>
      <Table><Row><Cell><Image alt="witta logo" src="http://hack2020.appmecha.co.uk/002-wisdom-sml.png"/></Cell>
      <Cell><Text>WITTA has not found any authority on this subject</Text></Cell></Row></Table>
    </InlineDialog> );
  }
}
```



ATLASSIAN

codegeist

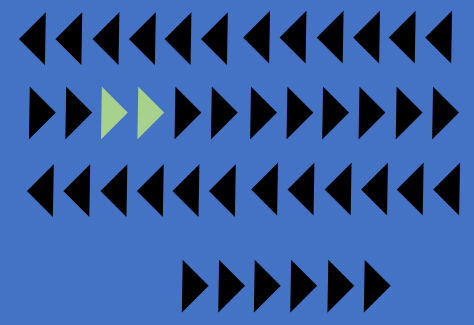
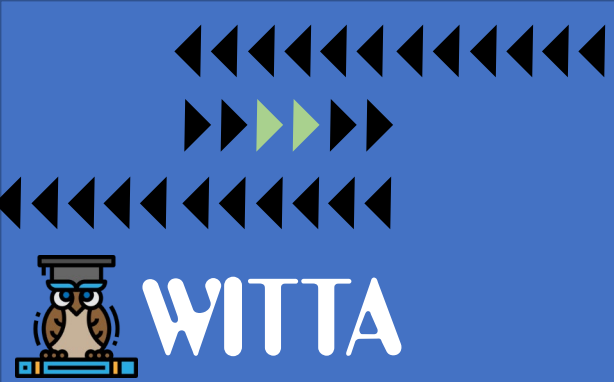


## Forge UI alert panels in JIRA

hack2020@neilhighley.com

```
import ForgeUI, { render, Fragment, Text, IssuePanel, Image, Table, Row, Cell } from '@forge/ui';
import UsersFound from './users'
const WittaResults=props=>{
  return(
    <Fragment>
      <Table>
        <Row><Cell><Image src="http://hack2020.appmecha.co.uk/002-wisdom-sm1.png"/></Cell>
        <Cell><Text>These fellows may know something about your task!</Text></Cell>
        <Cell><UsersFound/></Cell></Row>
      </Table>
      <Text>Icons made by [Nhor Phai](https://www.flaticon.com/authors/nhor-phai) from [www.FlatIcon.com](https://www.flaticon.com/)</Text>
    </Fragment>
  )
}
const App = () => {
  return (
    <Fragment><WittaResults/></Fragment>
  );
};

export const run = render(
  <IssuePanel>
    <App />
  </IssuePanel>
);
```



## WITTA Ingestion (Atlassian hook)

hack2020@neilhighley.com

```
export async function wittaUpdateRequest(resource, body) {
  let opts = {};
  let wittaApiKey="odshfdshciuehiuvfnvdhnkj";
  let wittaUpdateAPIURL="https://mzh822kz64.execute-api.us-east-2.amazonaws.com/default/witta-update"
  if (body) {
    opts = Object.assign({
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(body)
    });
  }
  opts.headers = Object.assign(opts.headers, {
    "x-api-key": `Bearer ${wittaApiKey}`
  });

  const response = await api.fetch(`${wittaUpdateAPIURL}${resource}`, opts);
  if (!response.ok) {
    const err = `Error invoking ${resource} (WITTA): ${response.status} ${response.statusText}`;
    error(err);
    throw new Error(err);
  }
  const responseBody = await response.json();
  debug(`Response from WITTA: ${JSON.stringify(responseBody)}`);
  return responseBody;
}
```



WITTA

ATLASSIAN

codegeist

hack2020@neilhighley.com



## AWS Ingestion API via Lambda

```

exports.handler = async (event) => {
  const uuid = {
    v1:()=>{return '00001';}
  }

  let resp=[];
  let date='2001-01-01';
  let guid=uuid.v1();
  let teamId=event['teamid'];
  let projectId=event['projectid'];
  let siteId=event['siteid'];
  let userId=event['userid'];
  let c=0;
  let t=event['data'].length;
  const processData=(data)=>{
    //upload to witta-engine ML
    c++;
    resp.push(guid+'-'+siteId+'-'+projectId+'-'+teamId+'-'+userId+' data item '+c+' of '+t+' from '+data['source']+'(REF:'+data['ref']+') ingested');
  }

  resp.push("WITTA-UPDATE");
  event['data'].forEach(d=>processData(d));
  resp.push(guid+'-'+date+' All ('+c+') Items of Data Processed, thankyou');
  const response = {
    statusCode: 200,
    body: JSON.stringify(resp),
  };
  return response;
};

```



WITTA

ATLASSIAN

codegeist

hack2020@neilhighley.com



## AWS Query API via Lambda

```

exports.handler = async (event) => {
  let content=event['content-by-importance'];
  let people=['5f03a0bf1a26ad0014e8bfe6','5efe82df3404690bae97dd5c','5f03a0bf1a26ad0014e8bfe6','5f03a0c0502ce1001dc740ae','5f03a0bf502ce1001dc740ab']

  const PopulateResponse=(query)=>{
    //This would call the WITTA ML algorithms, populated by WITTA-Update
    let selected=[];
    let num=Math.floor(Math.random()*people.length);
    let highest=0;
    let total=0;
    for(var i=0;i<num;i++){
      let person=people[Math.floor(Math.random()*people.length)];
      let score=Math.random();
      selected.push({"userref":person,"score":score});
      if(highest<score){highest=score;}
      total=total+score;
    }
    let average=total/num;
    return {'query':event['query'], "numadepts":num, "maxscore":highest, "averagescore":average, "adepts":selected};
  }

  let resp=PopulateResponse(event['query'])
  const response = {
    statusCode: 200,
    body: JSON.stringify(resp),
  };
  return response;
};

```



**WITTA**

[HTTPS://WITTA.ATLASSIAN.NET](https://witata.atlassian.net)

[HTTPS://hack2020.appmecha.co.uk](https://hack2020.appmecha.co.uk)

[hack2020@neilhighley.com](mailto:hack2020@neilhighley.com)