ZineOne Implementation Document

The below steps would help us to implement ZineOne tagging on the React store front applications like Emerald and Sapphire.

ZineOne helps us to monitor the events on the application like Homepage, Checkout etc.

It uses the Google tag Manager (GTM) data available in the React application to capture the events.

Note: The most important is to capture the current user/guest user data if the user doesn’t login into the application.

Currently the user data is created whenever user add any items into the cart and proceed to checkout.

To create a guest user when the home page loads, please follow the below steps.

1. Clone the v9.1.10.0\_base from the git sandbox repository.
2. Go to the file ***react-store/src/\_foundation/configs/userRequiredService.ts***
3. Place the below code

## export const userRequiredServices = ["cart", "checkout","espot"]

To include the Zine one tagging library in the react store application.

We would need the ZineOne server API key.

Reach out to the ZineOne Account manager to get the API key.

Once you get the above details please follow below steps.

1. Clone the v9.1.10.0\_base from the git sandbox repository.
2. Open the react-store/src/App.tsx file and place the below code.

//initiate the zineone instance.

/\*\*

\* import the zineone script

\*

\*/

const script = document.createElement("script");

script.async = true;

script.src = "https://cdn.zineone.com/cloud/latest/z1m.js";

script.onload=()=>{

setTimeout(() => {

console.log('Zineone Script loaded!!!')

//code for initializing the Zineone tagging

const w:any= window;

w.ZineOne?.initialize("SERVER\_API\_KEY")

}, 0);

} document.body.appendChild(script)

1. Once you paste the above code, open the browser developer tools and see the console.

You will see ZineOne tagging initialised in the console.

1. **Note**: Once you have done the above code in the App.tsx, there is no need to do changes in the other pages. As we have already configured google tag manager in our application, It uses that data for logging the events.