

ForgotPasswordScreen



```
this.instance.get  
<ForgotPasswordResponse>  
('auth/forgot-password?  
correo=' + email);
```

endpoint

user goes to email



gmail, outlook, hotmail deletes
any link that does not start with
'http' or 'https' in <a/>

endpoint

```
{http://192.168.1.70:4000}  
/api/auth/redirect/  
{0197495@up.edu.mx}
```

WEB_LINK

req.query.correo

option 1

option 2

http://192.168.1.70:4000/api/auth/redirect/:corr

endpoint



```
{spikyapp://changepassword}
/{0197495@up.edu.mx}
```

this is an html file rewritten
every time the server handles it,
rewritten with <a/> now.
file is in public/



```
this is an html file rewritten everytime the  
server handles it.  
window.onload = function() {  
  window.location.href =  
    "spikyapp://changepassword/0197495@up.edu.mx";  
}
```

```
let url = process.env.DEEP_LINK;  
let email = req.params.corr;  
let html = templateRedirectHTML(url+'/'+email);  
fs.writeFileSync("./public/mail.html", html);  
res.sendFile('mail.html', {})
```

```
route.get('/redirect/:corr',  
  handleRedirect);
```

endpoint

ChangePasswordScreen



```
return this.instance.put  
<UpdatePasswordUri>  
('auth/change-password-uri',{  
  email,  
  nuevaContrasena:  
    newPassword,});
```

endpoint

LoginScreen



await updatePasswordUri(...);
navigation.navigate('LoginScreen');

must be in
same stack

1. Create linking object for props in <NavigationContainer/>

```
const linking = {  
  prefixes: ['spikyapp://'],  
  config: {  
    initialRouteName: 'HomeScreen',  
    screens: {  
      HomeScreen: {  
        path: 'homescreen',  
      },  
      ChangeForgotPasswordScreen: {  
        path: 'changeforgotpassword/:correo',  
      },  
    },  
  },  
};
```

2. Using URI-scheme package to configure URI schemes:

```
$ npx uri-scheme add spikyapp --ios  
$ npx uri-scheme add spikyapp --android
```

3. Test the link from terminal:

```
$ npx uri-scheme open spikyapp://changeforgotpassword/someemail --android
```

4. To receive the correo param in ChangeForgotPasswordScreen:

```
export const ChangeForgotPasswordScreen = ({ route }) => { ... }
```

5. Configure all endpoints and screens following the previous diagram.