📑 RedwoodJS now with RSC

yarn rw g cell Sample


```
export const QUERY = gql`
  # GraphQL Query

`

export const Loading = () => <div>Loading...</div>

export const Empty = () => <div>Empty</div>

export const Failure = ({ error }) => (
  <div style={{ color: 'red' }}>Error: {error?.message}</div>
)

export const Success = ({ blank }) => {
  return <div>{JSON.stringify(blank)}</div>
}
```

```javascript
export const data = () => {
  return { products: db.products.findMany() }
}


export const Loading = () => <div>Loading...</div>


export const Empty = () => <div>Empty</div>


export const Failure = ({ error }) => (
  <div style={{ color: 'red' }}>Error: {error?.message}</div>
)


export const Success = ({ products }) => {
  return <div>{JSON.stringify(products)}</div>

}
```

```jsx
const Routes = () => {
  return (
    <Router useAuth={useAuth}>
      <Route path="/" page={HomePage} name="home" />
      {/* ------------- AUTH PAGES ------------------------------- */}
      <Set wrap={AuthLayout}>
        <Route path="/login" page={LoginPage} name="login" />
        <Route path="/signup" page={SignupPage} name="signup" />
        <Route path="/forgot-password" page={ForgotPasswordPage} name="forgotPassword" />
        <Route path="/reset-password" page={ResetPasswordPage} name="resetPassword" />
        <Route path="/accept" page={AcceptInvitePage} name="acceptInvite" />
      </Set>
      {/* ------------- LEGAL PAGES ------------------------------- */}
      <Set wrap={LegalLayout}>
        <Route path="/terms" page={TermsPage} name="terms" prerender />
```