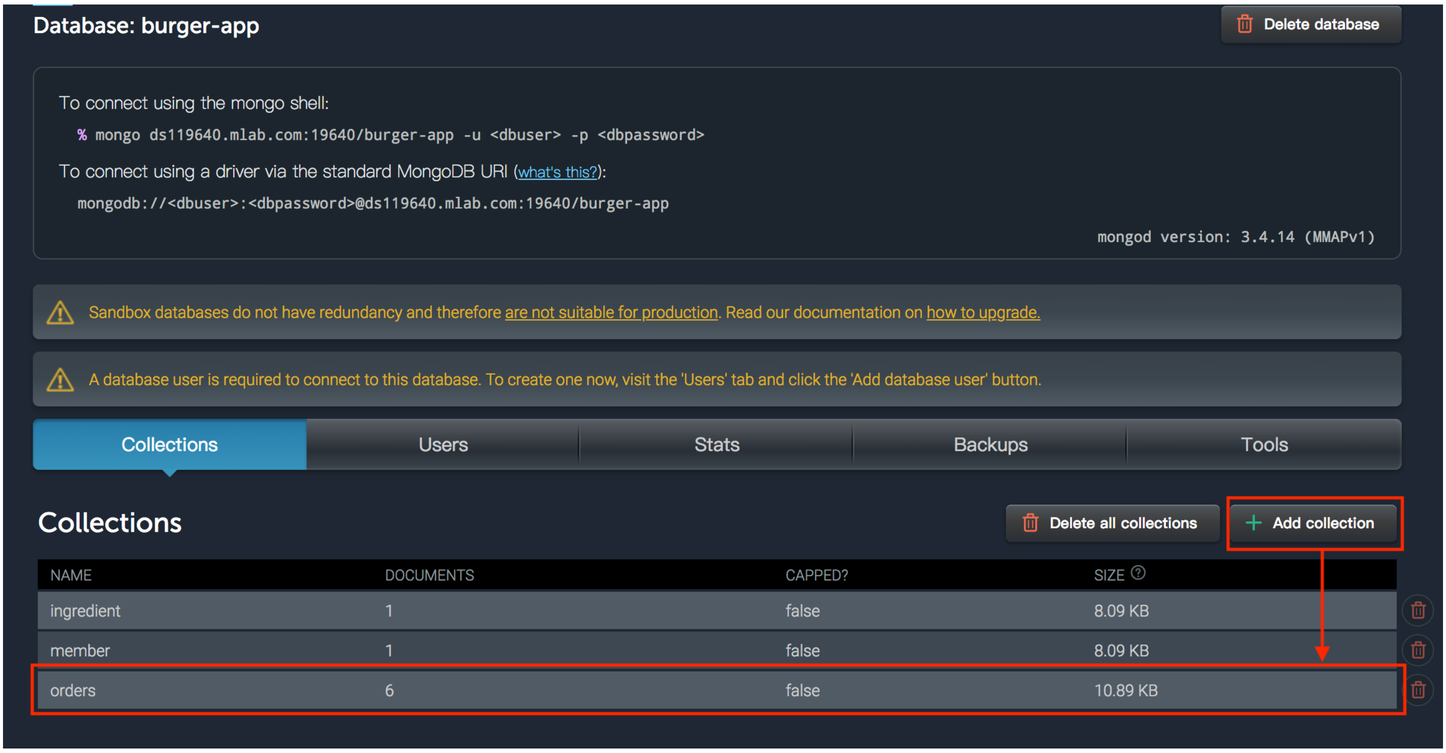
Burger-App 串接API運作訂單功能

在mLab的burger-app專案新增Collections orders



Database: burger-app

To connect using the mongo shell:

```
% mongo ds119640.mlab.com:19640/burger-app -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds119640.mlab.com:19640/burger-app
```

```
mongod version: 3.4.14 (MMAPv1)
```

⚠ Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on how to upgrade.

⚠ A database user is required to connect to this database. To create one now, visit the 'Users' tab and click the 'Add database user' button.

Collections

NAME	DOCUMENTS	CAPPED?	SIZE
ingredient	1	false	8.09 KB
member	1	false	8.09 KB
orders	6	false	10.89 KB

在src/store/action/order.js，將purchaseBurger跟fetchOrders串連API

本範例的orders api網址為 `https://api.mlab.com/api/1/databases/burger-app/collections/orders?apiKey=0CoMXpFAZiYEoelEnhF6Ve5HRt7sOrzL`

- 將漢堡訂單資訊送出至mlab

```
export const purchaseBurger = ( orderData, token ) => {
    return dispatch => {
        dispatch( purchaseBurgerStart() );
+       const url = "https://api.mlab.com/api/1/databases/burger-app/collections/orders?apiKey=0CoMXpFAZiYEo
-       axios.post( '/orders.json?auth=' + token, orderData )
+       axios.post( url, orderData )
            .then( response => {
                dispatch( purchaseBurgerSuccess( response.data.name, orderData ) );
            } )
            .catch( error => {
                dispatch( purchaseBurgerFail( error ) );
            } );
    };
};
```

- 從mlab獲取所有漢堡訂單資料

```
export const fetchOrders = (token, userId) => {
    return dispatch => {
        dispatch(fetchOrdersStart());
+       const url = "https://api.mlab.com/api/1/databases/burger-app/collections/orders?apiKey=0CoMXpFAZiYEo
+       axios.get( url )
-       const queryParams = '?auth=' + token + '&orderBy="userId"&equalTo="' + userId + '"';
-       axios.get( '/orders.json' + queryParams)
            .then( res => {
                const fetchedOrders = [];
                for ( let key in res.data ) {
                    fetchedOrders.push( {
                        ...res.data[key],
                        id: key
                    } );
                }
                dispatch(fetchOrdersSuccess(fetchedOrders));
            } )
            .catch( err => {
                dispatch(fetchOrdersFail(err));
            } );
    };
};
```

修改 src/App.js 在非登入狀態中提供checkout跟orders的路徑

```
  render () {
    let routes = (
      <Switch>
        <Route path="/auth" component={asyncAuth} />
        <Route path="/" exact component={BurgerBuilder} />
+       <Route path="/checkout" component={asyncCheckout} />
+       <Route path="/orders" component={asyncOrders} />
        <Redirect to="/" />
      </Switch>
    );

    if ( this.props.isAuthenticated ) {
      routes = (
        <Switch>
          <Route path="/checkout" component={asyncCheckout} />
          <Route path="/orders" component={asyncOrders} />
          <Route path="/logout" component={Logout} />
          <Route path="/auth" component={asyncAuth} />
          <Route path="/" exact component={BurgerBuilder} />
          <Redirect to="/" />
        </Switch>
      );
    }

    return (
      <div>
        <Layout>
          {routes}
        </Layout>
      </div>
    );
  }
}
```

修改 src/components/Navigation/NavigationItems/NavigationItems.js 顯示orders Item

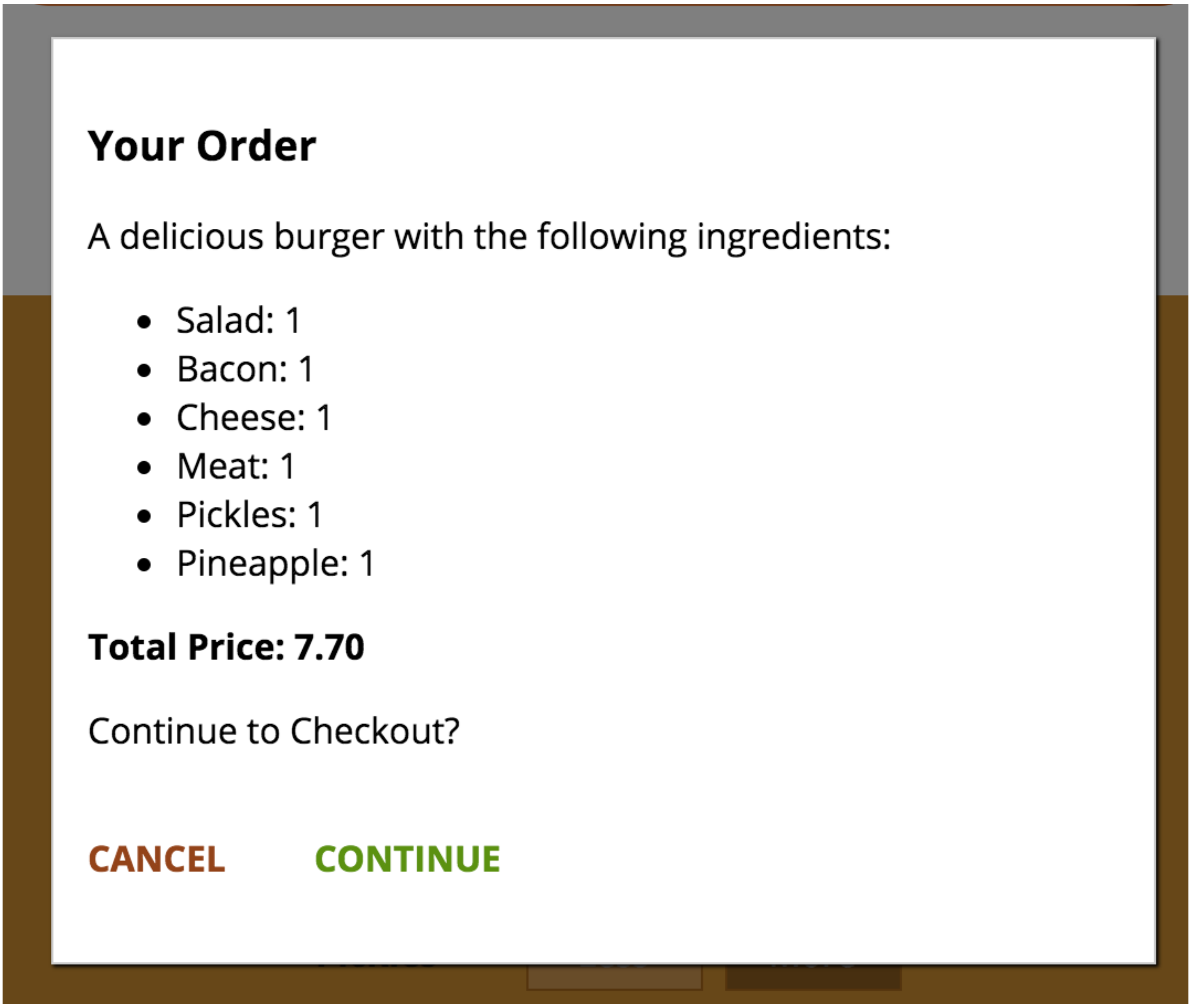
```
    const navigationItems = ( props ) => (
        <ul className={classes.NavigationItems}>
            <NavigationItem link="/" exact>Burger Builder</NavigationItem>
+           <NavigationItem link="/orders">Orders</NavigationItem>
-           {props.isAuthenticated ? <NavigationItem link="/orders">Orders</NavigationItem> : null}
            {!props.isAuthenticated
                ? <NavigationItem link="/auth">Authenticate</NavigationItem>
                : <NavigationItem link="/logout">Logout</NavigationItem>}
        </ul>
    );
```

修改 /src/containers/BurgerBuilder/BurgerBuilder.js 判斷是否跳出訂單結果

```
    purchaseHandler = () => {
+       this.setState( { purchasing: true } ) ;
-       if (this.props.isAuthenticated) {
-           this.setState( { purchasing: true } );
-       } else {
-           this.props.onSetAuthRedirectPath('/checkout');
-           this.props.history.push('/auth');
-       }
    }
```

實際運作

- 訂單結果



Your Order

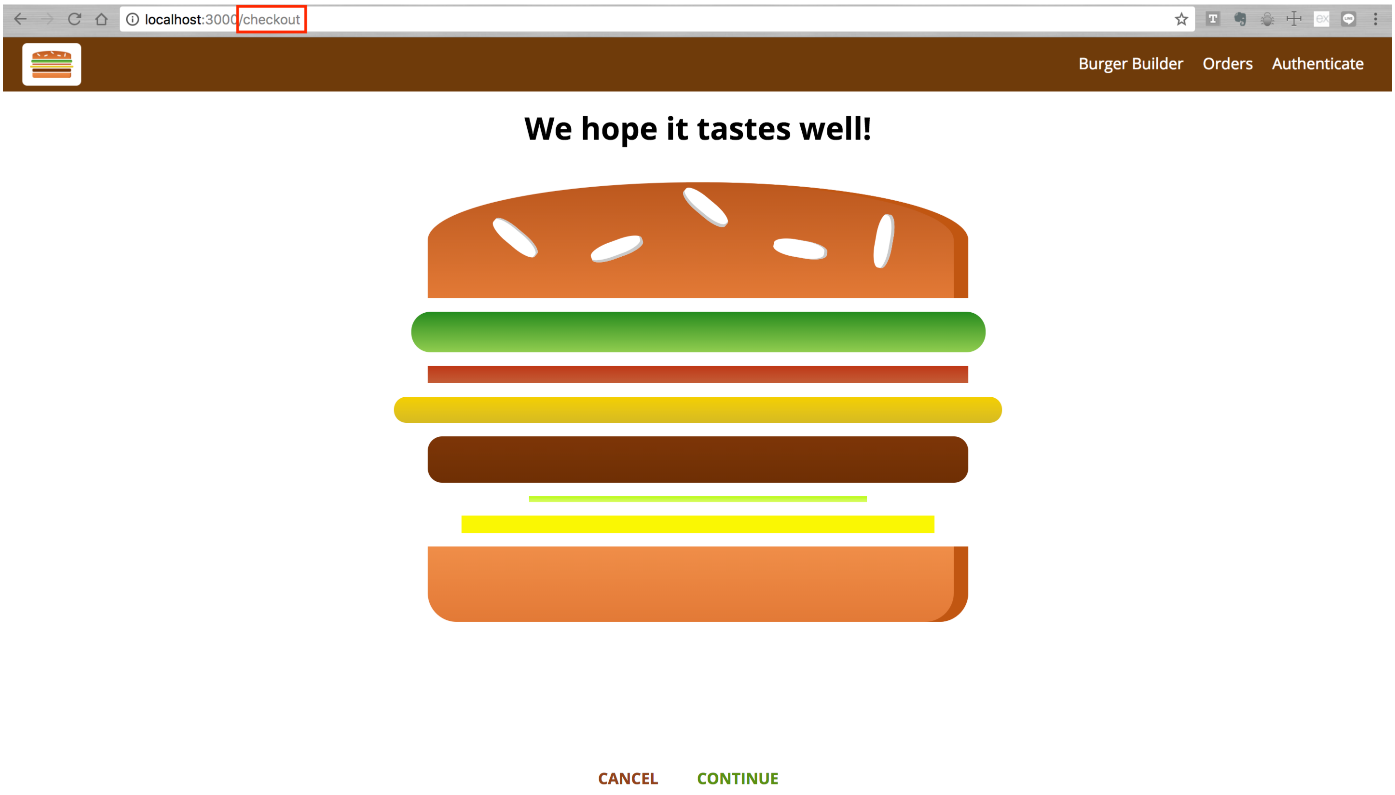
A delicious burger with the following ingredients:

- Salad: 1
- Bacon: 1
- Cheese: 1
- Meat: 1
- Pickles: 1
- Pineapple: 1

Total Price: 7.70

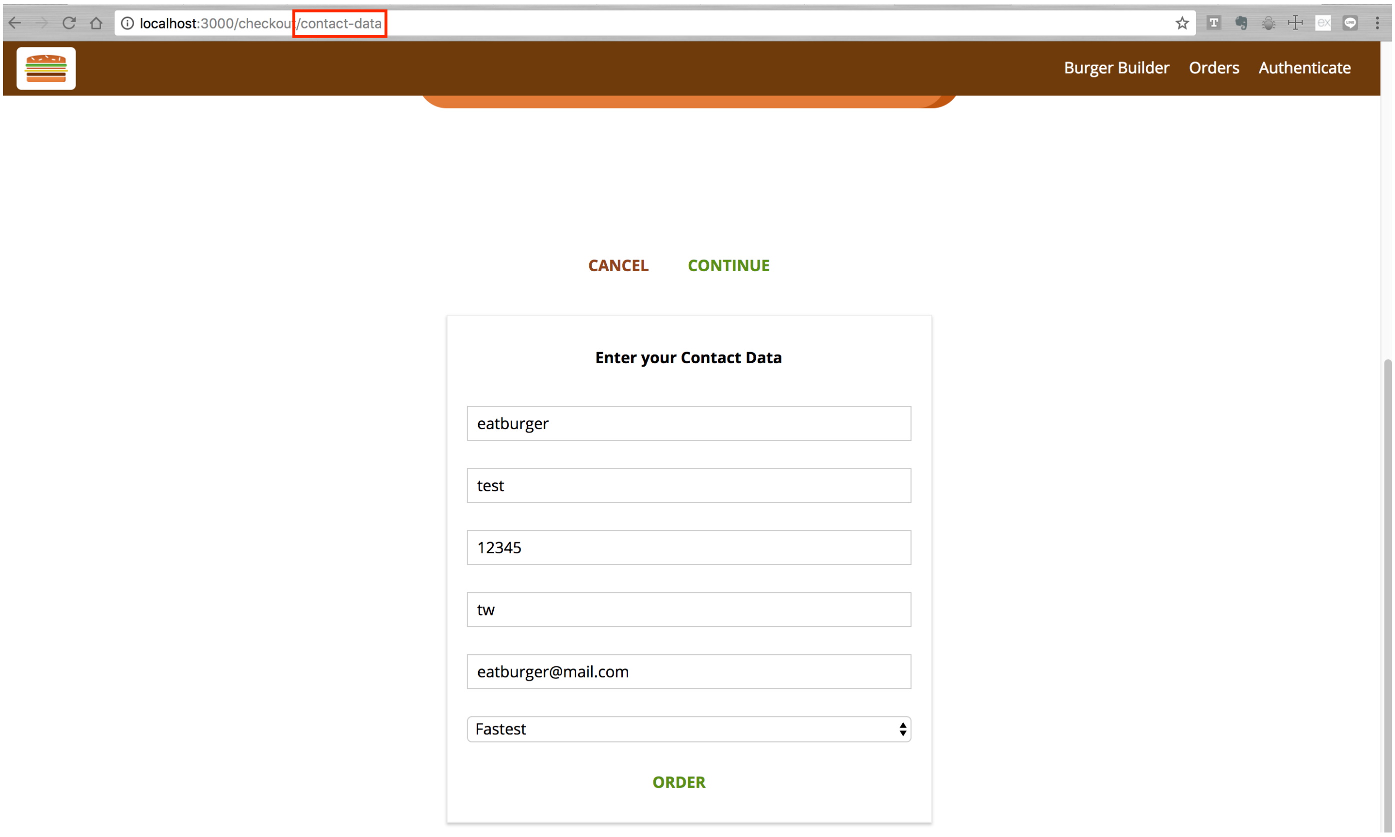
Continue to Checkout?

- checkout 頁面



We hope it tastes well!

- checkout/contact-data 頁面



Enter your Contact Data

eatburger

test

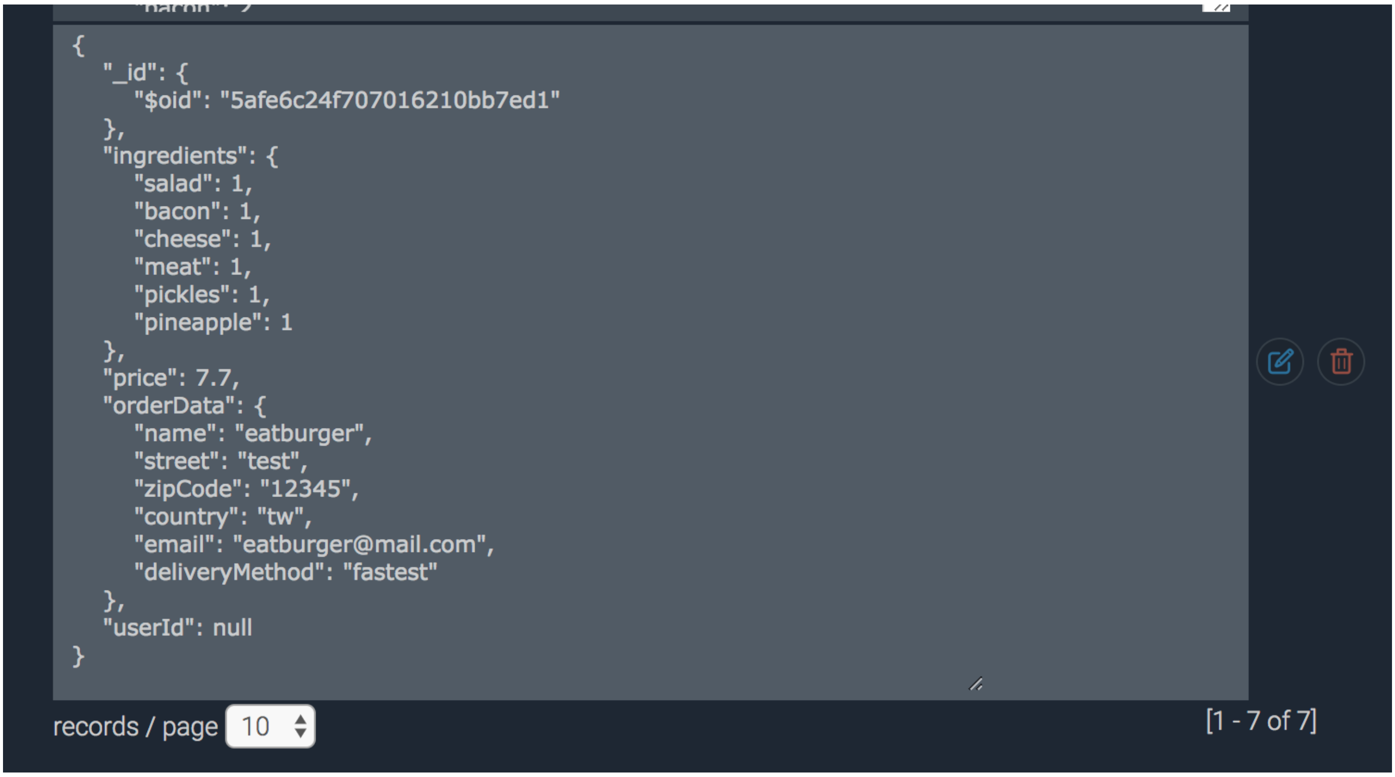
12345

tw

eatburger@mail.com

Fastest

- 到mLab的burger-app/orders 看是否最新訂單進來

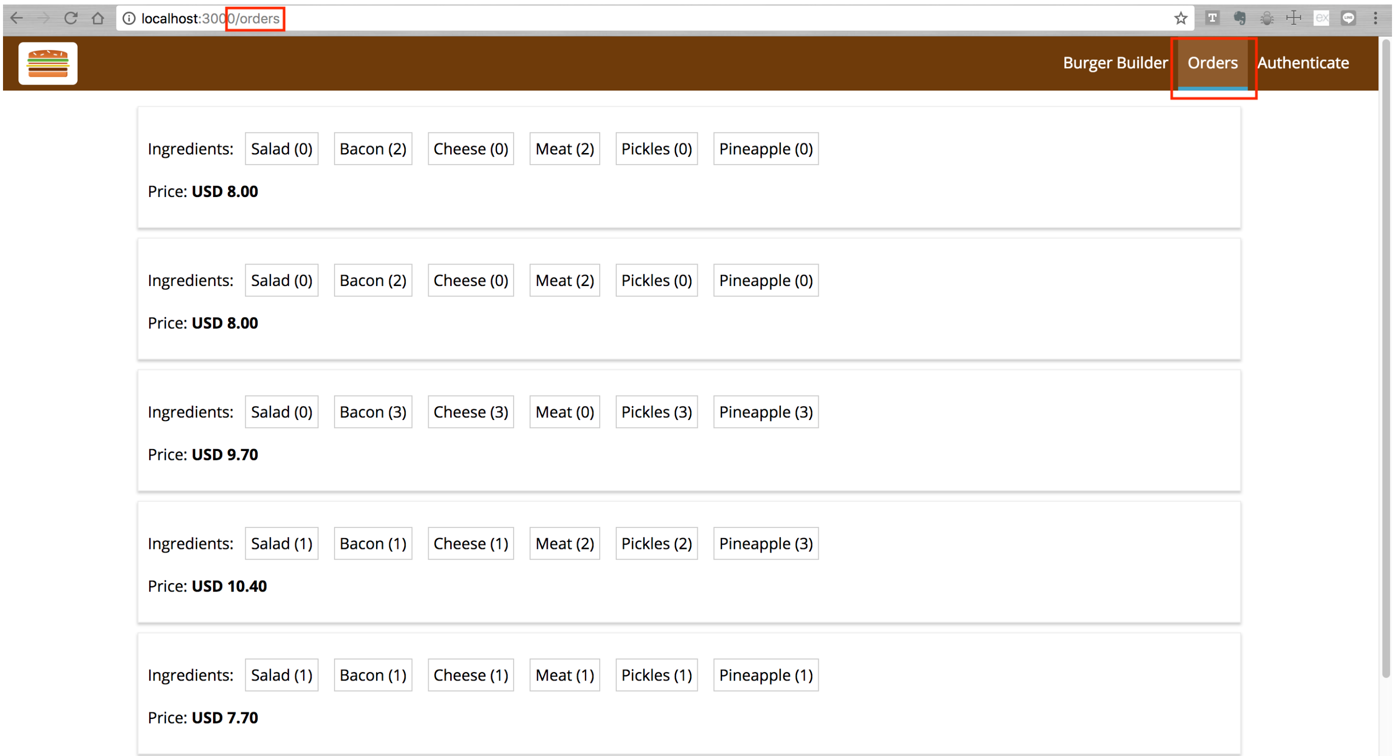


```
{
   "_id": {
      "$oid": "5afe6c24f707016210bb7ed1"
   },
   "ingredients": {
      "salad": 1,
      "bacon": 1,
      "cheese": 1,
      "meat": 1,
      "pickles": 1,
      "pineapple": 1
   },
   "price": 7.7,
   "orderData": {
      "name": "eatburger",
      "street": "test",
      "zipCode": "12345",
      "country": "tw",
      "email": "eatburger@mail.com",
      "deliveryMethod": "fastest"
   },
   "userId": null
}
```

records / page 10

[1 - 7 of 7]

- orders 頁面



Ingredients: Salad (0) Bacon (2) Cheese (0) Meat (2) Pickles (0) Pineapple (0)

Price: **USD 8.00**

Ingredients: Salad (0) Bacon (2) Cheese (0) Meat (2) Pickles (0) Pineapple (0)

Price: **USD 8.00**

Ingredients: Salad (0) Bacon (3) Cheese (3) Meat (0) Pickles (3) Pineapple (3)

Price: **USD 9.70**

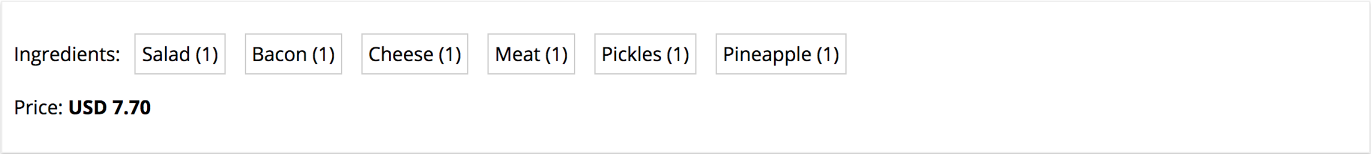
Ingredients: Salad (1) Bacon (1) Cheese (1) Meat (2) Pickles (2) Pineapple (3)

Price: **USD 10.40**

Ingredients: Salad (1) Bacon (1) Cheese (1) Meat (1) Pickles (1) Pineapple (1)

Price: **USD 7.70**

- 最新一筆訂單資訊



Ingredients: Salad (1) Bacon (1) Cheese (1) Meat (1) Pickles (1) Pineapple (1)

Price: **USD 7.70**