Email (Embedded Before)

This action occurs before a an email is sent. Changes made to the email's content and subject. The email can also be suppressed and not sent at all.

Action Type	Embedded
Full Action ID	embedded.commerce.email.render.before
Runs multiple custom functions?	No

Examples

Prefix Subject & Conditional Suppression

```
exports.handler = function(context, callback) {
 try {
   const subject = context.items.getSubject;
   const model = context.items.getModel;


   if (subject === 'order.changed' && !model?.items?.length) {
     context.items.exec('suppressEmail');
     return callback();
   }


   context.exec.setSubject('MyBrand ' + subject]);
   callback();
 } catch { callback(); }
};
```

Dynamic Template Swap & Model Enrichment

```
exports.handler = function(context, callback) {
 try {
   const subject = context.items.getSubject;
   const model = context.items.getModel;


   if (subject === 'order.changed' && model?.isCurbside) {
     context.exec.setTemplate('order.curbside.custom');
     context.exec.setSubject('Your curbside order is confirmed');
   }


   if (model?.items?.length) {
     model.highDuty = model.items.some(i => i.dutyAmount > 100);
     context.exec.setModel(model);
   }
   callback();
 } catch { callback(); }
};
```

Example injecting dynamic content using model

```
exports.handler = function (context, callback) {

    console.log('this is the test for changing the email content dynamically');

    // scenario: add custom content to order changed email


    var model = context.items.getModel;
    var user = context.items.getUser;
    var firstName = user.firstName;


    console.log('user first name is ' + firstName);
    console.log('order number is ' + model.orderNumber);
    console.log('email subject is ' + context.items.getSubject);


    var subject = context.items.getSubject;

    if (subject == "order.changed") {
        model.customContent = "This is custom content added dynamically through arc for user " + f
irstName + " for order " + model.orderNumber;
        context.exec.setModel(model);


        var newContent = context.items.getModel;
        console.log('new content is ' + JSON.stringify(newContent, null, 2));
        console.log('set new content successfully');
    }


    callback();
};
```