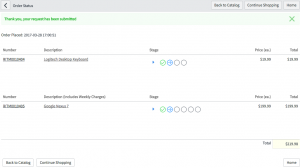
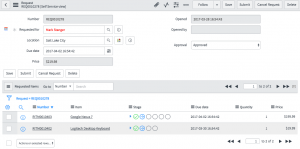
Overriding the ess Checkout View for Catalog Requests

ServiceNow’s service catalog interface is designed to give users a more familiar “Amazon.com-like” ordering experience. While this interface meets the needs of most customers, there are areas where certain modifications may be desirable. One of these areas may be the catalog checkout page. The out-of-box setup always presents every user with a specific checkout or ‘ess’ view for the ‘sc\_request’ (Request) table. This checkout page is forced every time any user attempts to view a request record using the ‘ess’ view.

[](https://www.servicenowguru.com/wp-content/uploads/2017/03/RequestDefaultView.png)

But what if you would rather use the standard ‘ess’ view that is similar to the rest of the forms in the system? Something like this…?  
[](https://www.servicenowguru.com/wp-content/uploads/2017/03/RequestOverrideView.png)

While this isn’t a typical modification, this article shows how you could accomplish it.

**This modification requires an override of out-of-box Service catalog functionality that may cause issues with later upgrades. You should make a note of any of these modified records so that you can restore them to their out-of-box state in the future if necessary. Please also note that while this modification may be useful in certain situations, the vast majority of ServiceNow implementations do not require it.**

1 Navigate to the ‘Navigation Handlers’ table by using the following URL (‘https://your-instance-name/sys\_navigator\_list.do’) or by typing ‘sys\_navigator.list’ in the left navigation search bar.

2 Open the ‘sc\_request’ record and change the ‘Table’ field value to ‘–None–‘. This effectively disables the navigation handler that overrides the ‘ess’ view for the ‘sc\_request’ table.

Next, you need to force the standard ‘ess’ view when a user sees the request record immediately after submitting a request.

3 Open the ‘com.glideapp.servicecatalog\_checkout\_view’ (‘com\_glideapp.servicecatalog\_checkout\_view\_v2’ for newer instances) UI page under ‘System UI -> UI pages’ and change the name to ‘com.glideapp.servicecatalog\_checkout\_view-OLD’ (‘com\_glideapp.servicecatalog\_checkout\_view\_v2-OLD’ for newer instances)

4 Create a new UI page with the following settings…

**UI page**  
Name: ‘com.glideapp.servicecatalog\_checkout\_view’ (‘com\_glideapp.servicecatalog\_checkout\_view\_v2’ for newer instances)  
Description: The final “checkout” screen you see in the service catalog after a successful order is placed.  
Note: This page is an override of the out-of-box UI page for checkout. It redirects the user from the out-of-box checkout page to a standard ‘ess’ view for the ‘sc\_request’ table if there is more than one item in the request and to the ‘ess’ view for the ‘sc\_req\_item’ table if there is only a single item in the request.

**<?xml** version='1.0' encoding='utf-8' **?>**  
**<j:jelly** trim='false' xmlns:j='jelly:core' xmlns:g='glide' xmlns:j2='null' xmlns:g2='null'**>**  
    **<g2:evaluate** jelly="true"**>**  
        var item\_id = '${sysparm\_sys\_id}';  
        var item\_table = 'sc\_request';  
                                                     
        var sc\_req\_item = new GlideRecordSecure('sc\_req\_item');  
        sc\_req\_item.addQuery('request', '${sysparm\_sys\_id}');  
        sc\_req\_item.query();  
        if(sc\_req\_item.getRowCount() == 1){  
            sc\_req\_item.next();  
            item\_table = 'sc\_req\_item';  
            item\_id = sc\_req\_item.sys\_id;  
        }  
    **</g2:evaluate>**  
    **<script** language='javascript'**>**  
        addLoadEvent(overrideView);  
        function overrideView() {      
            var url = '$[item\_table]' + '.do?sys\_id=' + '$[item\_id]' + '$[AMP]sysparm\_view=ess';  
            window.location = url;  
        }    
    **</script>**  
**</j:jelly>**

5 Verify the security for all of the elements and related lists on the new form. Since this view is not designed for use by ‘ess’ users out-of-box, you’ll need to make sure that the security on the request and request item tables meets the needs of your organization.

Overriding the Request Item (sc\_req\_item) ess view

The default behavior for users with no role is to force the ‘ess’ view for the Request Item table. If you find it necessary to override the ess view for the Request Item table you can modify or disable the ‘Catalog Scripts’ AJAX Script record. You can find this record by navigating to ‘System Definition -> AJAX Scripts’ if that module is active in your instance or you can enter ‘sys\_script\_ajax.list’ in your left navigation filter to navigate directly to the list without a module.