Payment References in Resurs Checkout

As of ☑ **RWC-256** - Use updateCheckoutOrderLines as safe layer during "desynched" cart DONE or Github issue #52, the way payment references are handled are changed. This explanation can also be found in the comments for the issue.

View the code change here.

How payment references are set in Resurs Checkout

This feature very much changed the structure of how orders are created, and this is also an important step to the "PUT" method in the RCO (which is the one that updates content in the cart on direct changes *on the checkout page*). This also affects

☑ **RWC-163** - Make sure the cart is always synchronizing in rco DONE which will only kick in when this feature don't work.

As this function is implemented I also had to fix the way payment id's work: As we need an active payment id in different sections of the plugin, the plugin also has to store the current payment id created in PHP for a longer time which now is controlled by payment id session aging:

When a customer enters the website with a valid cart, the plugin will check if there is an old "preferred payment id" to assign to the RCO iframe. If this exists in the session, it will reuse that id for up to an hour as long as defaults are used. This can be controlled by both the configuration (wp-admin) and a filter if adaptions has to be made.

When customers are successful, this payment id will be nulled and next time the customer returns to the page, a new payment id will be recreated and reused until the age is too high or it is expired by a successful payment. Failures is not included. On failures, the payment id will still be intact.

Also, as this feature is included we've fixed a potential updatePaymentReference issue with the payment id. Since we now know which ID we are using during the payment, updatePaymentReference will also affect the ID in the payment process and update the session based preferred ID. Since the order has already been created when we reach the updatePaymentReference section, we will tell the plugin that the new preferred id on next reload is the one already set during updatePaymentReference. This is a way to handle a plausible issue that yet is no traced by anyone, where some payment references are unexplanable lost.