# 如何向分析报告和 CSV 下载添加列

向分析报告添加列是一种非常有趣的方式，可以为 WooCommerce 添加功能。新的数据可以在用户界面的表格视图中以及用户最喜欢的电子表格或第三方应用程序中使用，通过生成 CSV 文件来实现。

这些说明假设您已安装并激活了一个 WooCommerce 测试插件。您可以按照 "入门" 指令 设置一个测试插件。该帖子还包括进一步修改用于获取数据的查询的说明，以便以更高级的方式进行操作 - 仅添加一个简单的列并非必需。

在 WooCommerce 中，分析 CSV 文件有两种不同的生成方式：一种是在 Web 浏览器中使用已下载的数据，另一种是在服务器上使用新的查询。它使用数据集的大小来确定方法 - 如果结果超过一页，则会在服务器上生成数据，并通过电子邮件向用户发送链接；如果结果适合一页，则会在浏览器中直接生成并下载数据。

我们将首先介绍在服务器上添加列的方法，因为这是生成发送到浏览器的数据的过程。

此示例扩展了 “下载 ” 分析报告。为了在您的系统中获取一些用于此报告的数据，请创建一个可下载的产品，并设置下载过期值，然后创建一个购买该产品的订单，并多次下载该产品。在测试中，我创建了 26 次下载，这足以使报告在每页显示 25 个项目时分布在两页，而在每页显示 50 个项目时分布在一页。这使我能够测试在服务器和浏览器中生成的 CSV 文件。

在您的插件的 PHP 代码中，添加三个过滤器处理程序:

```
// 此代码将 SELECT 片段添加到报告 SQL 中
function add_access_expires_select( $report_columns, $context, $table_name ) {
□if ( $context !== 'downloads' ) {
□□return $report_columns;
□}
□$report_columns['access_expires'] =
□□'product_permissions.access_expires AS access_expires';
□return $report_columns;
}
add_filter( 'woocommerce_admin_report_columns', 'add_access_expires_select', 10, 3 );

// 此代码将列标题添加到 CSV 文件中
```

```
function add_column_header( $export_columns ) {
$export_columns['access_expires'] = 'Access expires';
return $export_columns;
}
add_filter( 'woocommerce_filter_downloads_export_columns', 'add_column_header' );

// 此代码将查询到的项目映射到导出的项目
function map_access_expires( $export_item, $item ) {
$export_item['access_expires'] = $item['access_expires'];
return $export_item;
}
add_filter( 'woocommerce_report_downloads_prepare_export_item', 'map_access_expires', 10, 2 );
```

这会将访问过期时间戳添加到 “下载 ”表 /CSV 文件中（当 CSV 文件在服务器上生成时）。

这些三个过滤器共同作用，将新的列添加到数据库查询中，将新的标题添加到 CSV 文件中，并将从数据库返回的数据映射到 CSV 文件。第一个过滤器 `woocommerce_admin_report_columns` 向用于数据查询的 `SELECT` 语句添加一个 SQL 片段。第二个过滤器 `woocommerce_filter_downloads_export_columns` 向服务器生成的 CSV 文件添加列标题。第三个过滤器 `woocommerce_report_downloads_prepare_export_item` 将从数据库查询返回的数据 `$item` 中的值映射到 CSV 文件的导出项。

为了添加对在浏览器中生成的列的支持，还需要将另一个过滤器添加到您的插件的 JavaScript 代码中：

```
import { addFilter } from "@wordpress/hooks";
function addAccessExpiresToDownloadsReport(reportTableData) {
  const { endpoint, items } = reportTableData;
  if ("downloads" !== endpoint) {
    return reportTableData;
  }

  reportTableData.headers = [
    ...reportTableData.headers,
    {
      label: "访问过期时间",
      key: "access_expires",
    },
  ];
  reportTableData.rows = reportTableData.rows.map((row, index) => {
    const item = items.data[index];
    const newRow = [
```

```
      ...row,
      {
        display: item.access_expires,
        value: item.access_expires,
      },
    ];
    return newRow;
  });

  return reportTableData;
}
addFilter(
  "woocommerce_admin_report_table",
  "dev-blog-example",
  addAccessExpiresToDownloadsReport
);
```

此过滤器首先向 CSV 文件添加标题，然后映射数据。

使用您创建的插件，您现在应该能够将数据添加到分析表、服务器生成的 CSV 文件以及浏览器生成的 CSV 文件中。

---

版本号 #2
由 壳壳虫 创建于 2026-03-09 11:53:58 UTC
由 壳壳虫 更新于 2026-04-03 09:34:26 UTC