# XFS4IoT SP-Dev Workgroup

6 July 2021

## To add others from your company

Please email us at:

xfs4iot_sp-dev_info@kal.com

## XFS4IoT framework available: Card Reader

- Card Reader class **available** on GitHub since 4 May 2021
- Members can create a GitHub fork and start developing card reader SPs

## The next step in XFS4IoT SP-Dev: Cash Dispenser

- KAL targeted the second SP-Dev framework to be available on GitHub today. ***It is now available***.
  - Cash Dispenser class
- The initial release does not include:
  - End-to-end security
  - Cash recycling

## Review of 1 June SP-Dev Workgroup meeting

- **Discussed NuGet packages and how to use them**
  - — NuGet is a Package Manager for .NET providing tools to create, publish and consume packages
  - — Accessible directly from Visual Studio
  - — Enables developers to share reusable code
  - — Free and open-source, developed by Microsoft
  - — No need to download, fork or copy the framework source code
- **Examined sample framework code on GitHub**
- **Reviewed support for small devices and low bandwidth IoT connections**

ATM Software

# COMPETITION LAW COMPLIANCE GUIDELINES

Summary for the XFS4IoT SP-Dev Workgroup

## COMPETITION LAW COMPLIANCE GUIDELINES

- **We must not** agree to fix prices, co-ordinate bids, divide up territories, or agree not to compete.
- **We must not** discuss pricing strategy, terms and conditions of a deal, business strategy, deals won or lost, relationships with customers or partners, or share details of our company's technologies.
- **We must not** share competitively sensitive market information.
- **It is ok to** continue discussions on the development and adoption of XFS4IoT-based SPs.
- **We must** make access to workgroup product available on fair, reasonable and non-discriminatory terms.

ATM Software

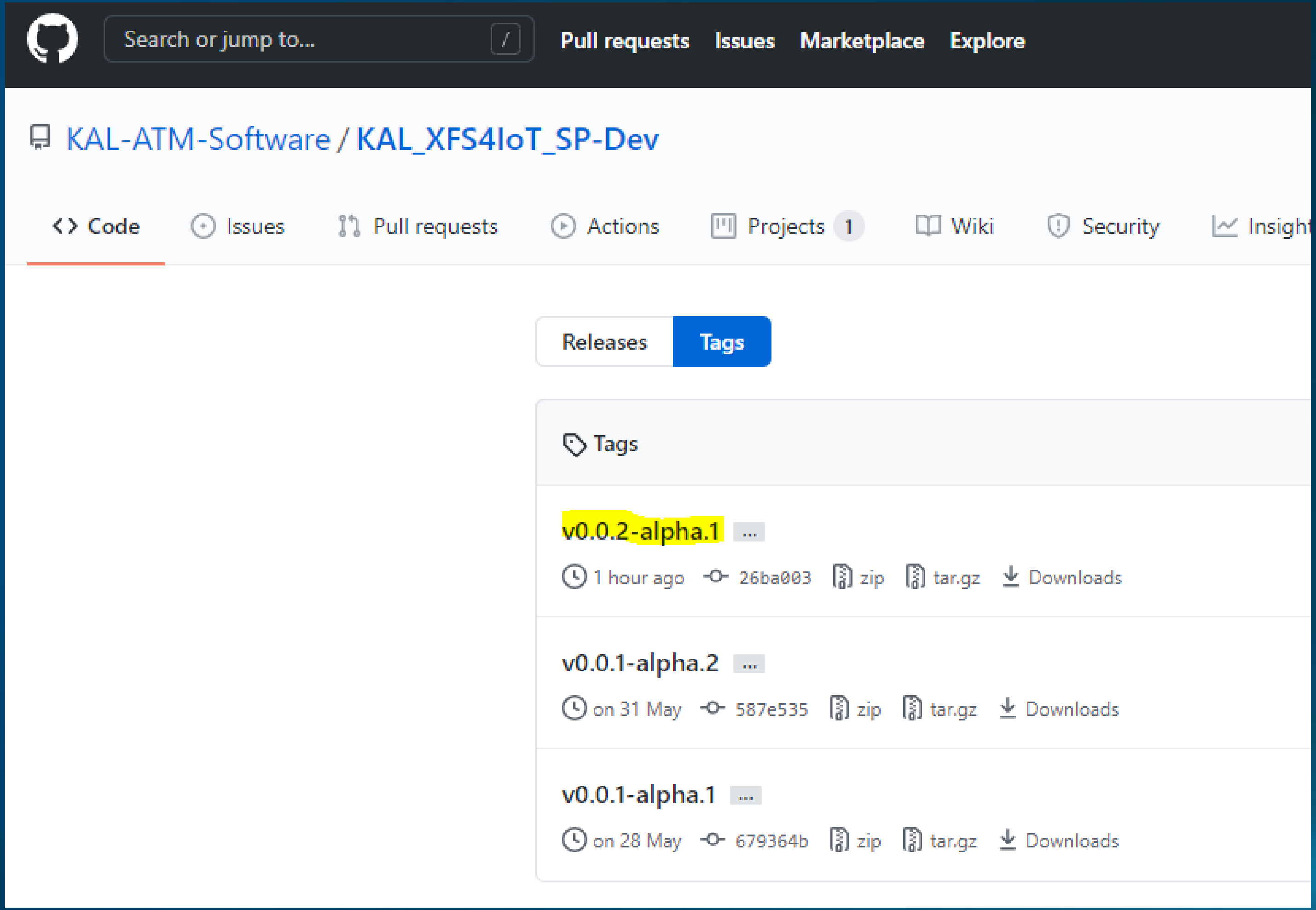
## Dispenser framework release

## Cash dispenser framework

- Available **now** in “**KAL_XFS4IoT_SP-Dev**” repo
- All Dispenser commands are supported
- New end-to-end security feature is **not** yet included
- All framework code is available to:
  - — Write test tools
  - — Implement XFS4 SPs
  - — Review
  - — Test with our sample

# Cash dispenser framework

- NuGet packages available
- Pre-release v0.0.2-alpha.1



KAL-ATM-Software / KAL_XFS4IoT_SP-Dev

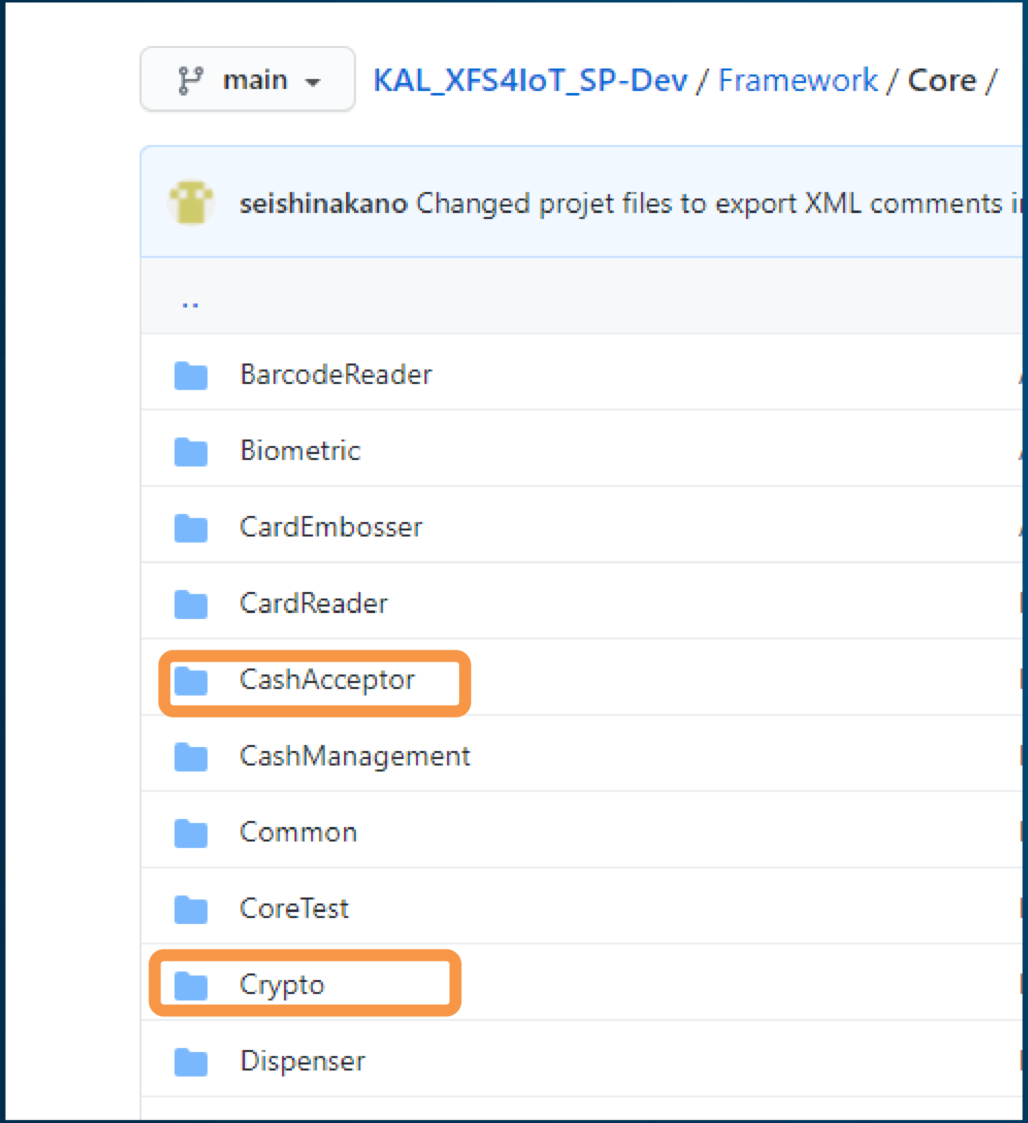
Tags

v0.0.2-alpha.1 ...
1 hour ago 26ba003 zip tar.gz Downloads

v0.0.1-alpha.2 ...
on 31 May 587e535 zip tar.gz Downloads

v0.0.1-alpha.1 ...
on 28 May 679364b zip tar.gz Downloads

# Cash dispenser framework



main KAL_XFS4IoT_SP-Dev / Framework / Core /

seishinakano Changed projet files to export XML comments i

..

BarcodeReader

Biometric

CardEmbosser

CardReader

CashAcceptor

CashManagement

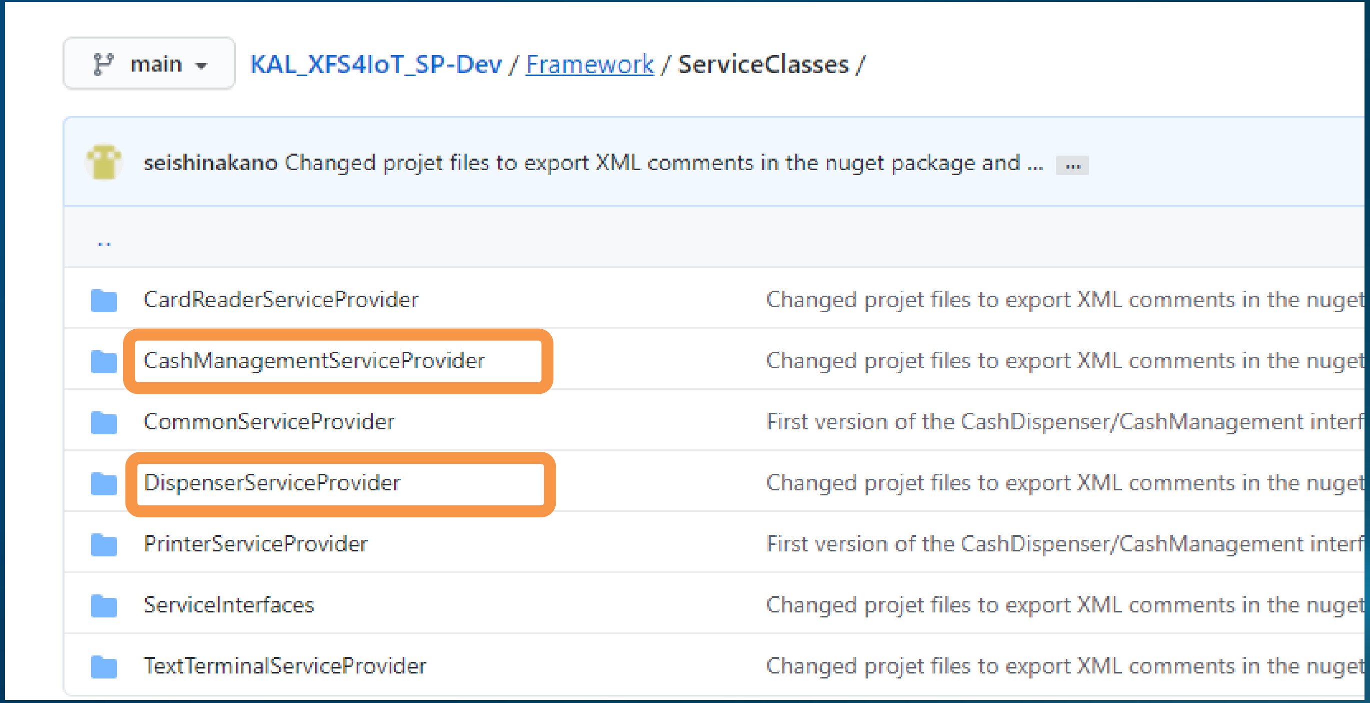
Common

CoreTest

Crypto

Dispenser

- Main components
  - Dispenser
  - Cash Management



main KAL_XFS4IoT_SP-Dev / Framework / ServiceClasses /

seishinakano Changed projet files to export XML comments in the nuget package and ... …

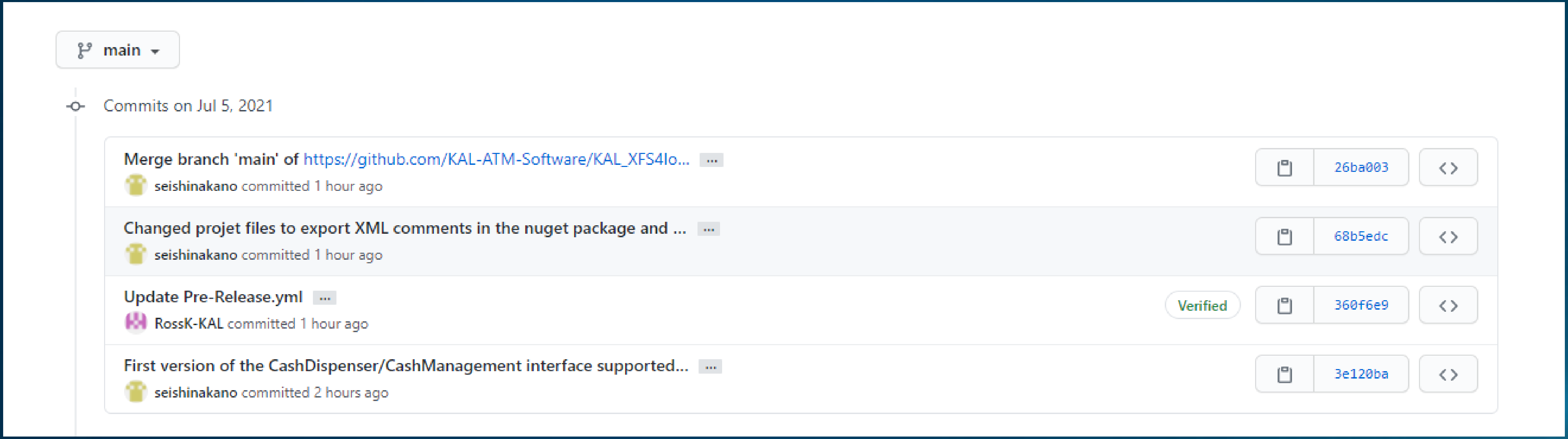
..

| | |
|---|---|
| CardReaderServiceProvider | Changed projet files to export XML comments in the nuget |
| CashManagementServiceProvider | Changed projet files to export XML comments in the nuget |
| CommonServiceProvider | First version of the CashDispenser/CashManagement interf |
| DispenserServiceProvider | Changed projet files to export XML comments in the nuget |
| PrinterServiceProvider | First version of the CashDispenser/CashManagement interf |
| ServiceInterfaces | Changed projet files to export XML comments in the nuget |
| TextTerminalServiceProvider | Changed projet files to export XML comments in the nuget |

# Cash dispenser framework

- Follow changes through Commit details
- All commits history available on GitHub:

  https://github.com/KAL-ATM-Software/KAL_XFS4IoT_SP-Dev/commits/main


main

Commits on Jul 5, 2021

Merge branch 'main' of https://github.com/KAL-ATM-Software/KAL_XFS4Io... …
seishinakano committed 1 hour ago — 26ba003

Changed projet files to export XML comments in the nuget package and ... …
seishinakano committed 1 hour ago — 68b5edc

Update Pre-Release.yml …
RossK-KAL committed 1 hour ago — Verified — 360f6e9

First version of the CashDispenser/CashManagement interface supported... …
seishinakano committed 2 hours ago — 3e120ba

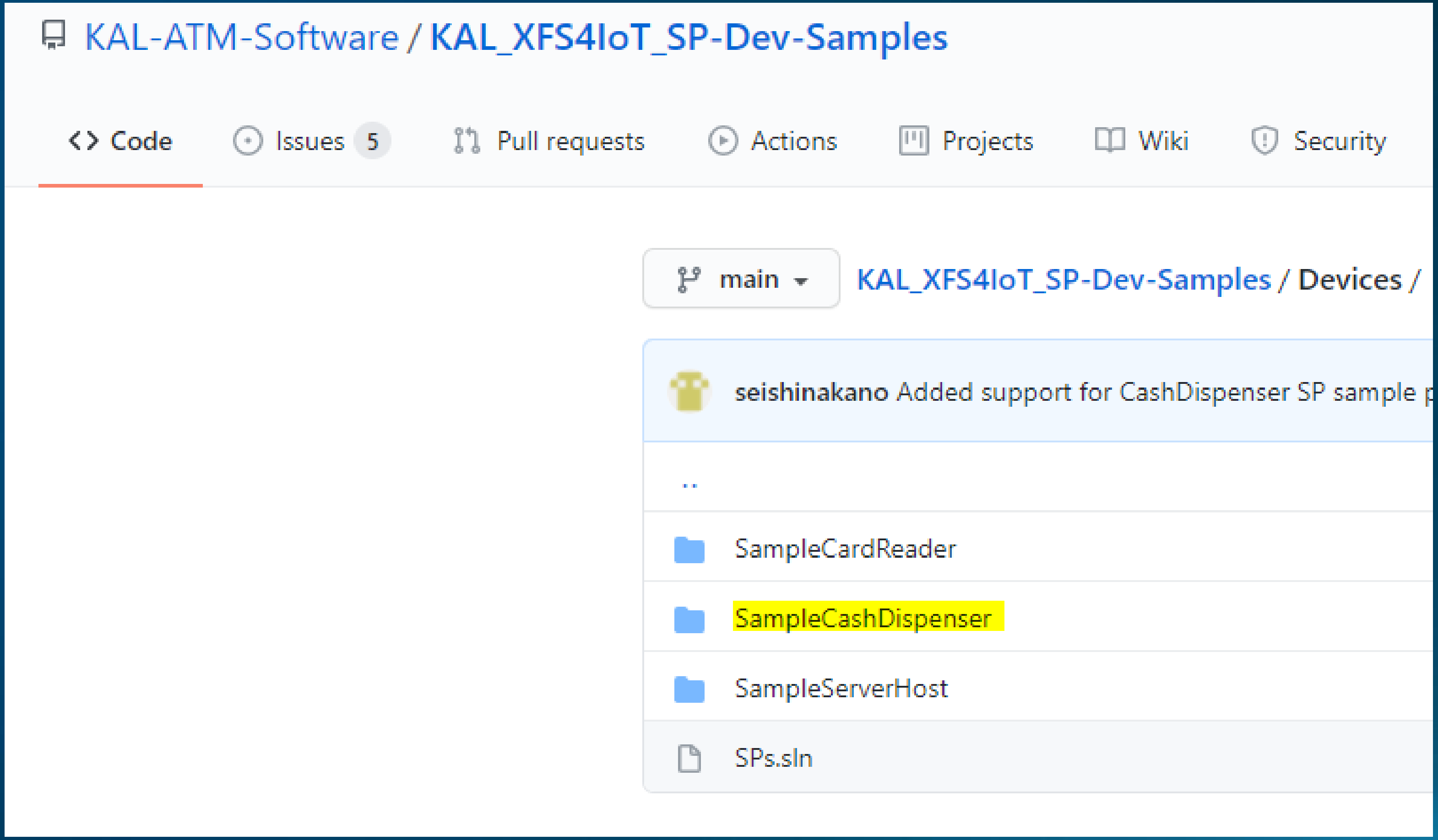
## Cash dispenser sample

- Available in “KAL_XFS4IoT_SP-Dev-Samples” repo
- Simple file structure – separated from the framework core code
- All that is needed to start writing an SP
- Supports all main commands:

  → Dispense, Present, Retract, Reject …

# Cash dispenser sample

Available in Sample repo


KAL-ATM-Software / KAL_XFS4IoT_SP-Dev-Samples

KAL_XFS4IoT_SP-Dev-Samples / Devices /

seishinakano Added support for CashDispenser SP sample p

..

SampleCardReader

SampleCashDispenser

SampleServerHost

SPs.sln

ATM Software

## Demo on real cash dispenser

## Cash dispenser demo

- Video will be made available on YouTube
- Link will be sent to all workgroup members via email

ATM Software

# Development language options

Current options for developing with the SP-Dev Framework

## What language?

Possible options:

- C99/C11/C++ – Popular for embedded programming. Low resource. Difficult to work with
- C#/.net – Easy and productive language
- Go, Rust – Modern safe languages
- …

## Current choice

C#9/.NET 5 (/6)

- Quick easy development
- Cross platform support for Windows + Linux
- Good fit for PC hardware and also ‘ARM SoC’ style systems

Note: .net nanoframework isn’t currently supported

## Other options

C99/C11/C++

- Would support very small hardware
- MCU. Memory measured in ‘K’, battery life measured in months
- KAL are considering this for the future…

Other languages:

- No current plans, but…

## Is there a compromise?

What about if we want to write customisation code in C++

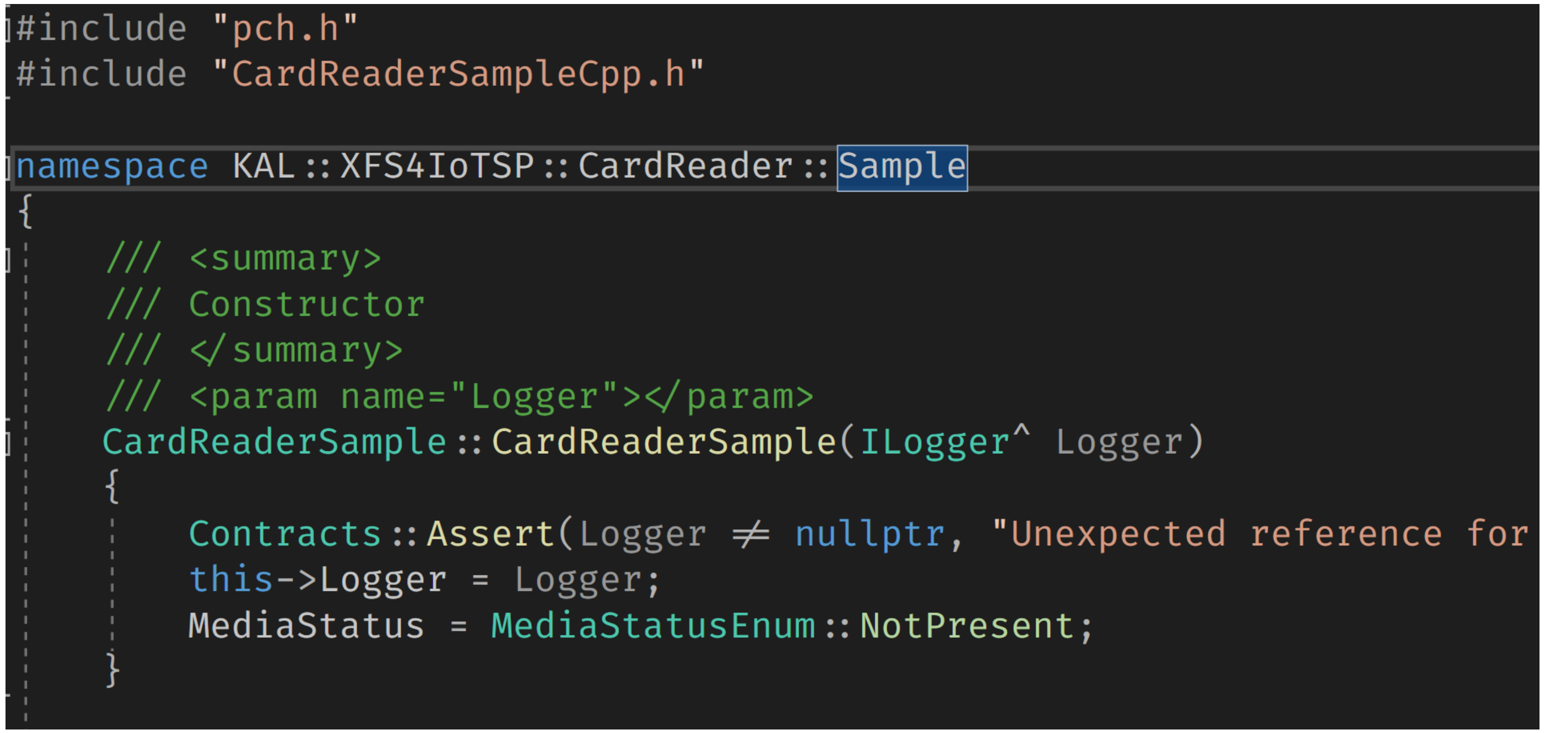
C# can call into native code. Two options:

- On Windows, C++ with /CLR support
- pInvoke on any platform

KAL have been experimenting with /CLR

# C++/CLR

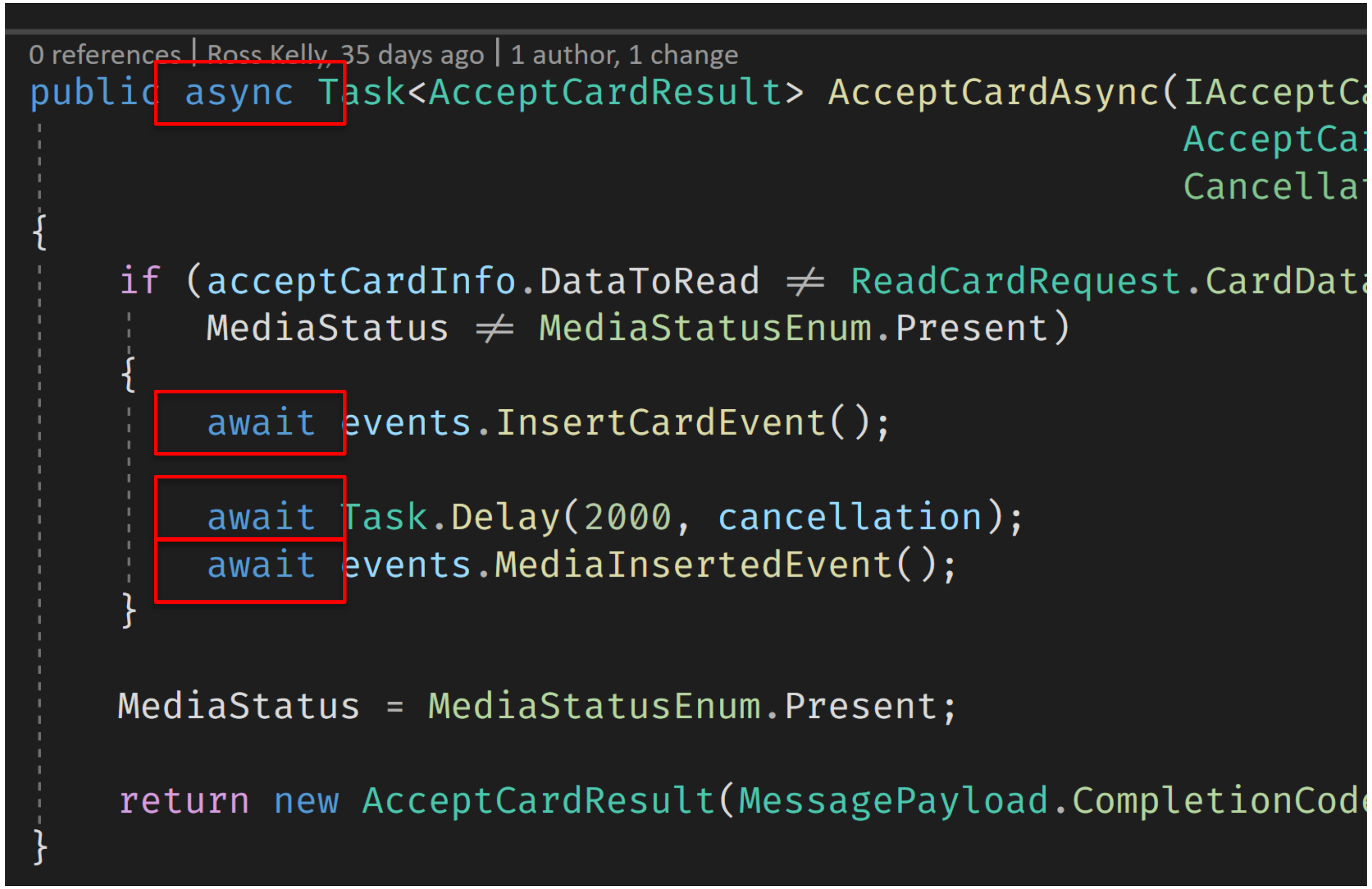
- /CLR makes it easy to mix C++ with .NET code
- “It just works” with ^ for references to .net types



```
#include "pch.h"
#include "CardReaderSampleCpp.h"

namespace KAL::XFS4IoTSP::CardReader::Sample
{
    /// <summary>
    /// Constructor
    /// </summary>
    /// <param name="Logger"></param>
    CardReaderSample::CardReaderSample(ILogger^ Logger)
    {
        Contracts::Assert(Logger ≠ nullptr, "Unexpected reference for
        this->Logger = Logger;
        MediaStatus = MediaStatusEnum::NotPresent;
    }
```

# Async code with C++

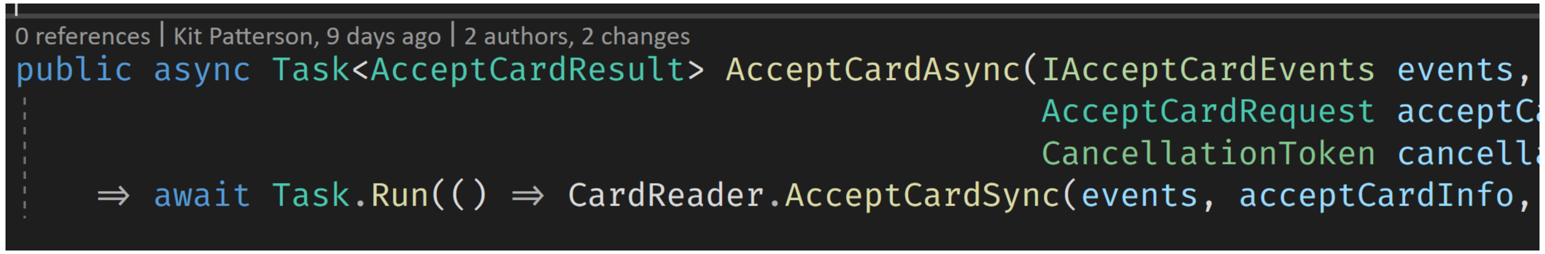
- SP-Dev framework uses async/await model
- Doesn't mix well with C++/CLR …



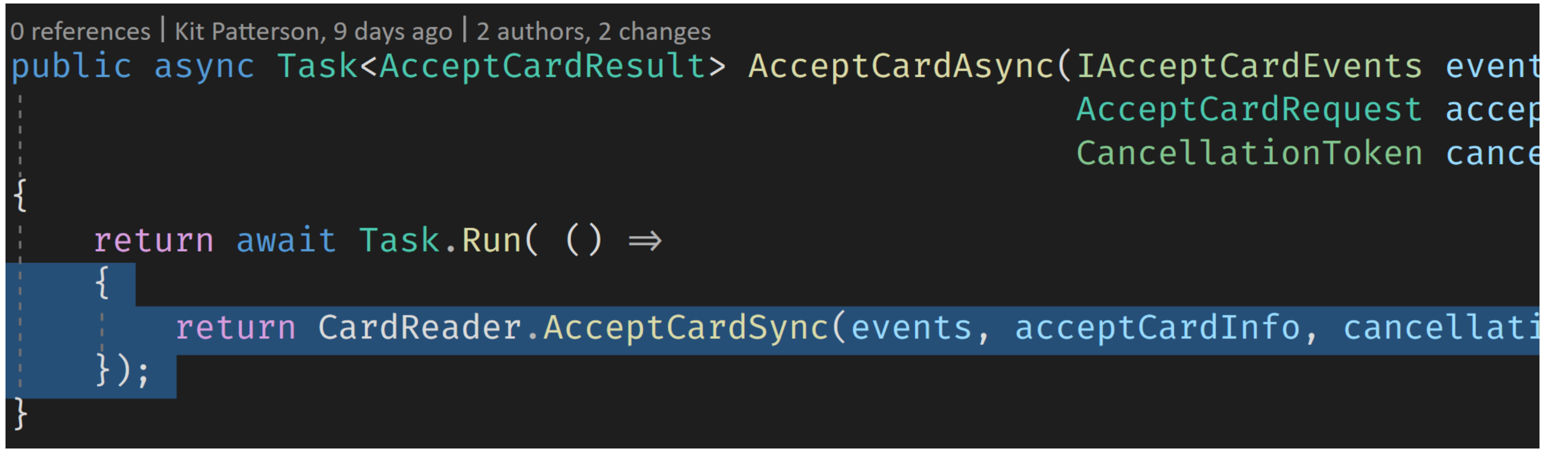
```
0 references | Ross Kelly, 35 days ago | 1 author, 1 change
public async Task<AcceptCardResult> AcceptCardAsync(IAcceptCa
                                                    AcceptCa
                                                    Cancella

{
    if (acceptCardInfo.DataToRead ≠ ReadCardRequest.CardData
        MediaStatus ≠ MediaStatusEnum.Present)
    {
        await events.InsertCardEvent();

        await Task.Delay(2000, cancellation);
        await events.MediaInsertedEvent();
    }

    MediaStatus = MediaStatusEnum.Present;

    return new AcceptCardResult(MessagePayload.CompletionCode
}
```

# Tast.Run wrapper

- Add a wrapper in C# to run the C++ code on a different thread
- Thread pool, so fast and low resources



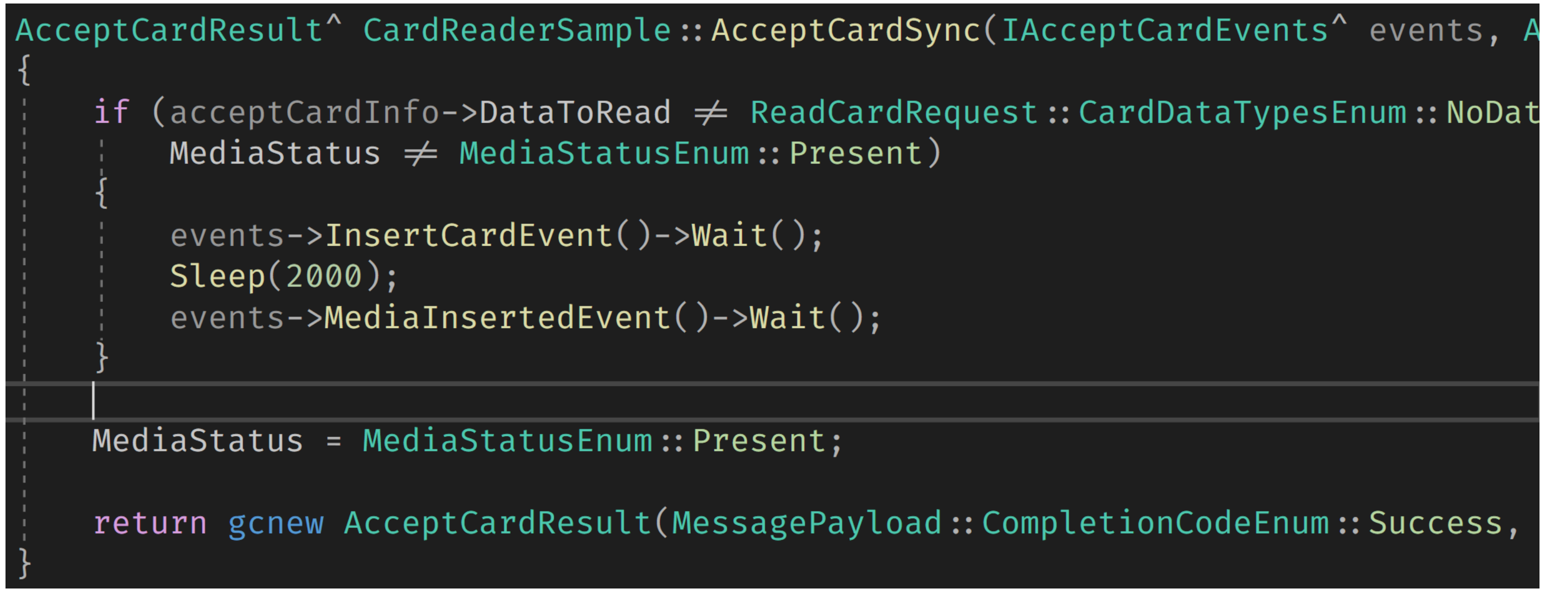
```
0 references | Kit Patterson, 9 days ago | 2 authors, 2 changes
public async Task<AcceptCardResult> AcceptCardAsync(IAcceptCardEvents events,
                                                     AcceptCardRequest acceptCa
                                                     CancellationToken cancella
    => await Task.Run(() => CardReader.AcceptCardSync(events, acceptCardInfo,
```



```
0 references | Kit Patterson, 9 days ago | 2 authors, 2 changes
public async Task<AcceptCardResult> AcceptCardAsync(IAcceptCardEvents event
                                                     AcceptCardRequest accep
                                                     CancellationToken cance
{
    return await Task.Run( () =>
    {
        return CardReader.AcceptCardSync(events, acceptCardInfo, cancellati
    });
}
```

# C++/CLR

- C++/CLR is now simple linear code
- C++/CLR code can now block



```
AcceptCardResult^ CardReaderSample::AcceptCardSync(IAcceptCardEvents^ events, A
{
    if (acceptCardInfo->DataToRead ≠ ReadCardRequest::CardDataTypesEnum::NoDat
        MediaStatus ≠ MediaStatusEnum::Present)
    {
        events->InsertCardEvent()->Wait();
        Sleep(2000);
        events->MediaInsertedEvent()->Wait();
    }

    MediaStatus = MediaStatusEnum::Present;

    return gcnew AcceptCardResult(MessagePayload::CompletionCodeEnum::Success,
}
```

## Conclusions

- Mixing C++/CLR is possible and even easy
- Similar code will work with pInvoke on Linux
- pInvoke will work for any language, not just C++

- Be careful with threads – code Task.Run code runs on a separate thread

ATM Software

## Text Terminal (TTU) framework

## Text Terminal framework

- Initial version *will be* available soon
- Public repo will be updated regularly
- Watch/Star to get all updates
- Release will be ready before August workgroup meeting

*Interest raised by workgroup members, we hope to have feedback!*

## Text Terminal framework

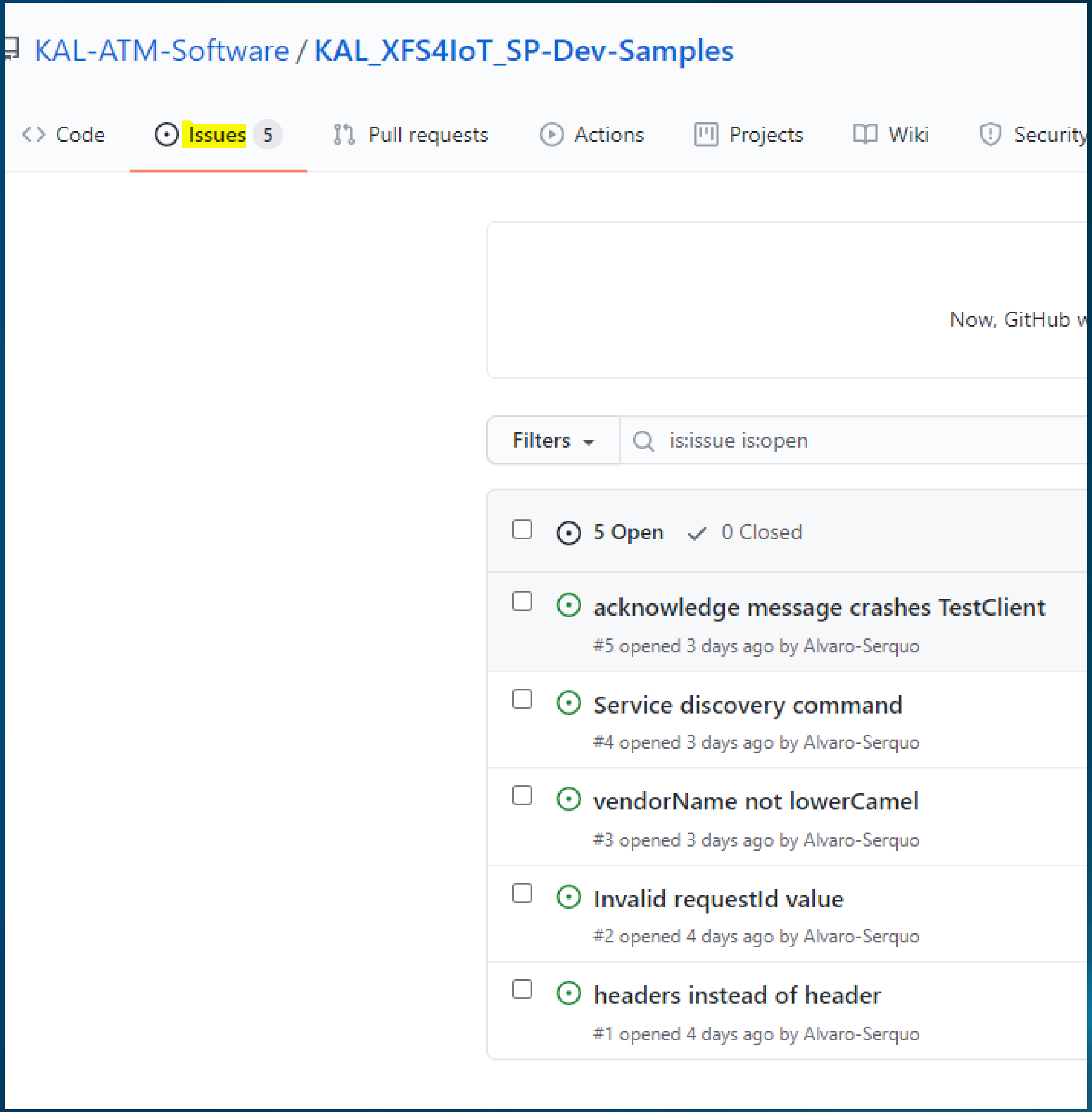
- Many commands will be supported in initial version:
  - Read
  - Write
  - Reset
  - ClearScreen
  - Beep
  - SetResolution and more…
- Sample code will be made available

ATM Software

## Issues on GitHub

# Issues on GitHub

- Issues were raised on GitHub
- Sample repo
- Element of response provided
- Action to fix taken
- Discussion
- Completely public!!



KAL-ATM-Software / **KAL_XFS4IoT_SP-Dev-Samples**

Code | Issues 5 | Pull requests | Actions | Projects | Wiki | Security

Now, GitHub w

Filters | is:issue is:open

5 Open ✓ 0 Closed

- **acknowledge message crashes TestClient**
  #5 opened 3 days ago by Alvaro-Serquo
- **Service discovery command**
  #4 opened 3 days ago by Alvaro-Serquo
- **vendorName not lowerCamel**
  #3 opened 3 days ago by Alvaro-Serquo
- **Invalid requestId value**
  #2 opened 4 days ago by Alvaro-Serquo
- **headers instead of header**
  #1 opened 4 days ago by Alvaro-Serquo

## Next call

MS Teams

- First Tuesday of each month at 1300 UK time

**Next call: 3rd August 2021, 1300 UK, 0800 US EST, 2100 Tokyo time**