

# Demo Proposal: Making Web Applications -XSafe

Amit Levy, David Terei, Deian Stefan, David Mazières

Stanford

{levya, dtere, deian, \_}@cs.stanford.edu

## Abstract

Simple is a web framework for Haskell. Simple came out of our work on Hails, a platform for secure web applications. For Hails, we needed a flexible web framework that uses no unsafe language features and can be used to build apps outside the IO monad. Unlike many mainstream web frameworks, Simple does not enforce a particular structure or paradigm. Instead, it simply provides a set of composable building blocks to help developers structure and organize their web applications.

We've used Simple to build both traditional web applications as well as applications with explicit, strong safety and security guarantees. In the demonstration, we'll focus on the former – introducing the framework and motivating its utility for traditional web apps – and show how we can leverage the LIO information flow control library to add mandatory security policies to apps.

**Keywords** Web application, Safe Haskell, Hails

## 1. Proposal

We propose to demonstrate *Simple*, a Haskell web framework with two key features:

1. *Simple* uses exclusively safe language features, notably avoiding tempting functionality such as Template Haskell.
2. IO is not hard-coded, and can be replaced with monads offering richer features such as LIO and STM.

*Simple* comes out of our work building Hails – a web platform that enforces security policies on applications using decentralized information flow control (DIFC). Hails (and the underlying DIFC library, LIO) relies on the Safe Haskell extension to ensure that untrusted apps do not violate the type system, and thus circumvent security policies. As a result, any web framework for Hails apps must be able to compile with -XSafe and must not run in the IO monad. Unfortunately existing Haskell web frameworks failed to satisfy both these points. They usually hardcode the IO monad, making it virtually impossible to repurpose them for Hails. Even worse, either the core framework or many of the supporting libraries use unsafe features of Haskell such as Template Haskell.

The result of this work, *Simple* is a general web application framework, useful for both traditional apps (in the IO monad) as well as apps written in constrained monads such as LIO and STM. By carefully designing the framework's abstractions with these constraints in mind, *Simple* achieves monad generality without sacrificing the power and flexibility that mainstream frameworks offer. Specifically, *Simple* is built around a single ControllerT monad:

```
newtype ControllerT s m a = Controller {  
  runController :: s -> Request ->  
    m (Either Response a, s)  
}
```

Almost all of the framework functionality, including routing, parsing forms, rendering templates and cookie-based sessions is agnostic of the underlying monad. The few exceptions (i.e., storing an uploaded file to /tmp) can be trivially rewritten for the target monad, and are merely library functions.

In the demonstration, we'll show how to build a traditional web application using *Simple* and how to add mandatory security policies by replacing the IO monad with LIO. We believe this flexibility of moving from IO to LIO with relative ease will help ease the adoption of secure application design with platforms like Hails.