DreamFactory Security Guide

This white paper is designed to provide security information about DreamFactory. The sections below discuss the inherently secure characteristics of the platform and the explicit security features that have been engineered into the software.

Native Install

DreamFactory is an open source software package that runs in the cloud or on premises. Single-click installers are available for most of the Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) cloud vendors. You can also run DreamFactory on a Linux, Windows, or Mac desktop computer. The software package includes a complete xAMP Stack application built with PHP including several default services (i.e. a dedicated SQL and NoSQL database, file storage, email, etc.).

These versatile installation options enhance security. A company can use the same secure deployment practices for DreamFactory that they already use for their other applications. You can monitor usage, make backups, control cost, optimize performance, configure firewalls, and deploy applications with familiar tools and systems. Companies can choose a trusted cloud vendor or their own data center to manage application security.

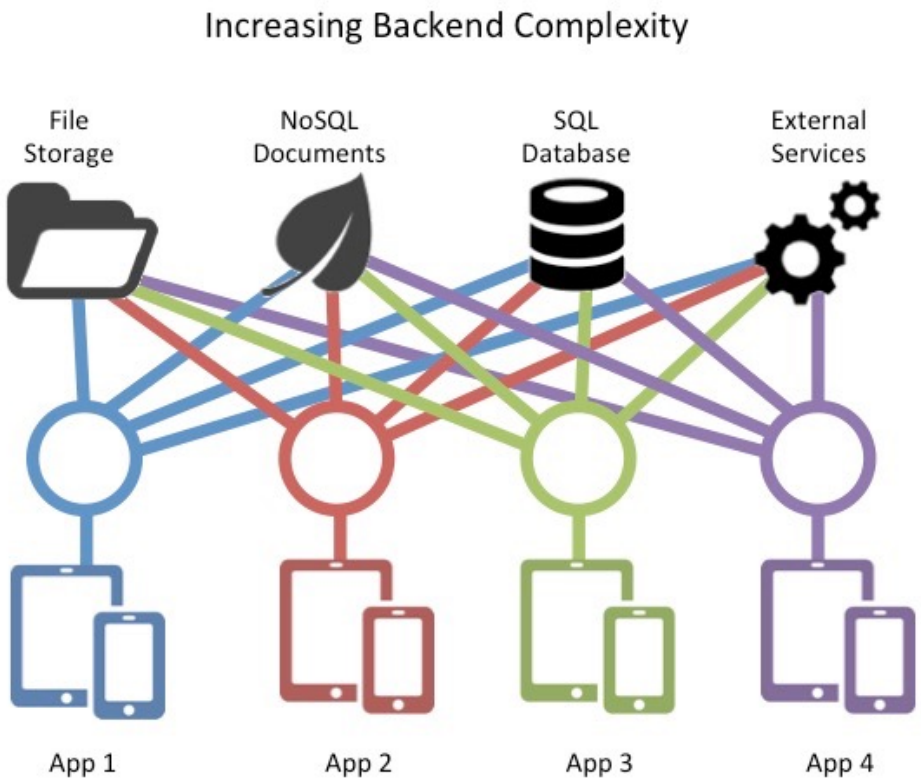
Open Source

The DreamFactory source code is available under the Apache License at GitHub. An open source product has "many eyes" on the code base. DreamFactory Software has signed up for third-party security audits as well. DreamFactory Software has quality assurance engineers looking for security problems on a regular basis. Thousands of independent developers are also testing the platform daily. The open source nature of the code base enhances security.

The latest version of the software is available on various cloud marketplaces and can be installed on your infrastructure. Since DreamFactory Software doesn't host your application, there is no risk of our company losing your data. You do not need to worry about where our data center is located, who has access to your private information, or how we keep your data separate from other customer accounts.

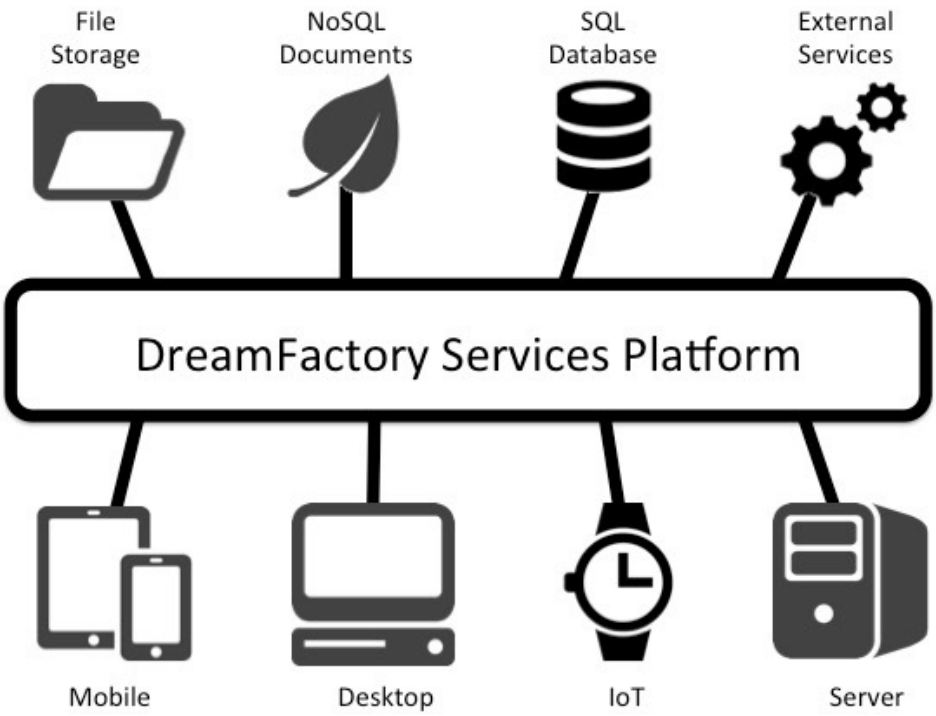
Automatically Generated Services

REST API complexity is a major security challenge. With each new mobile project, there's a tendency to create more and more REST APIs. Each service might be written in house, by different developers, or by external consultants. These services typically have different user management systems, different security protocols, different parameter styles, and different requests and responses. They will be hardwired to various data sources, and designed to run on different pieces of physical infrastructure. Over time, the server-side infrastructure can become increasingly complex, and each new REST API endpoint can potentially introduce new security vulnerabilities.



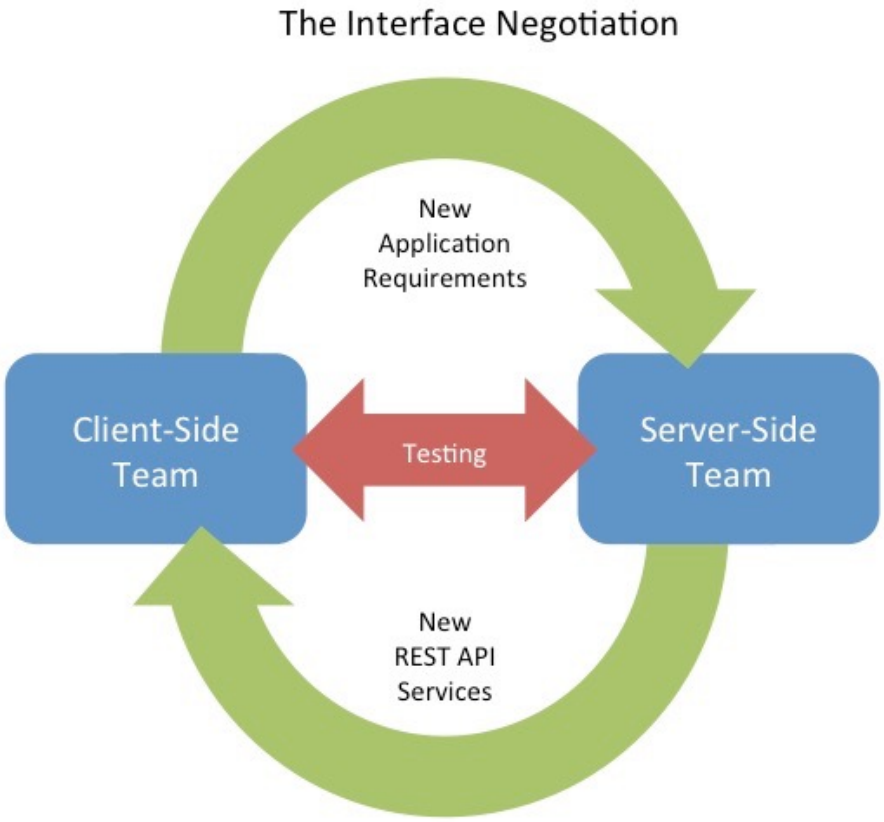
DreamFactory provides a comprehensive and reusable palette of REST API services. All of the various backend data sources are accessed through a unified REST API interface. This provides a reusable platform of REST API services for general-purpose application development. The total number of exposed service endpoints is reduced, and the connections to backend data sources are standardized.

Reusable, General-Purpose REST API Platform



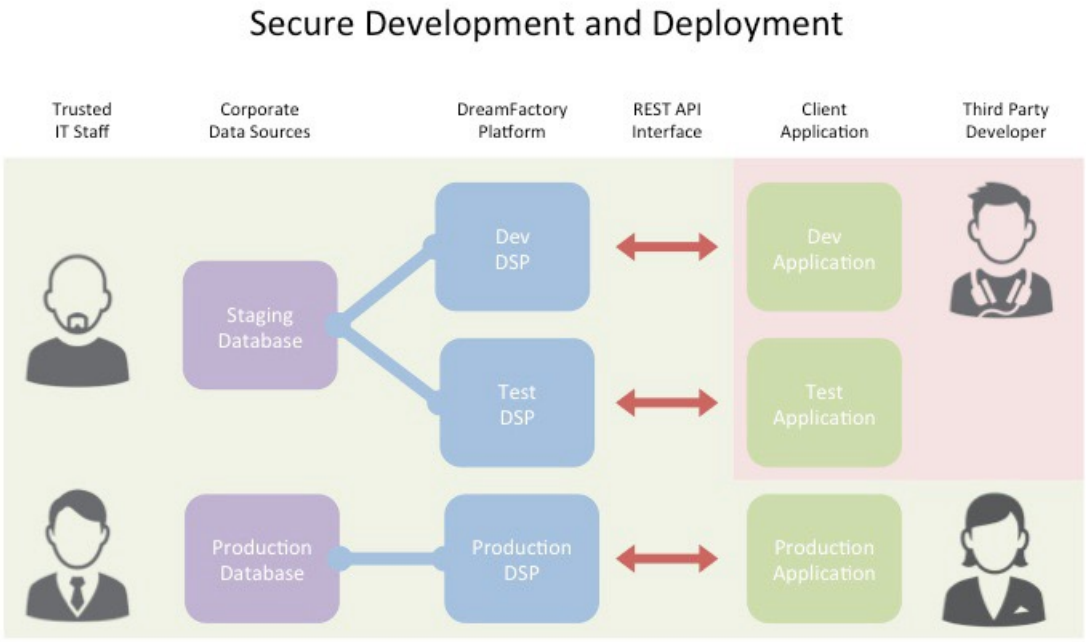
Service Platform

In a typical mobile development project, there is a client team and a server team, and the two groups must agree on a REST API interface. DreamFactory stops this back and forth interface negotiation and allows each side to focus on their respective responsibilities. DreamFactory enhances security by decoupling client-side application development from server-side user management and administration.



External Partners

In many mobile projects, external consultants create the client-side application and also build the server-side interface. With DreamFactory, trusted members of the IT staff can easily create a REST API for each new mobile project. The platform can provide access to a staging or production database as needed. The client application can be moved from development, to testing, and on to production without code changes. This strategy eliminates the need to expose corporate information and backend systems to external partners working on development projects.



Service Abstraction

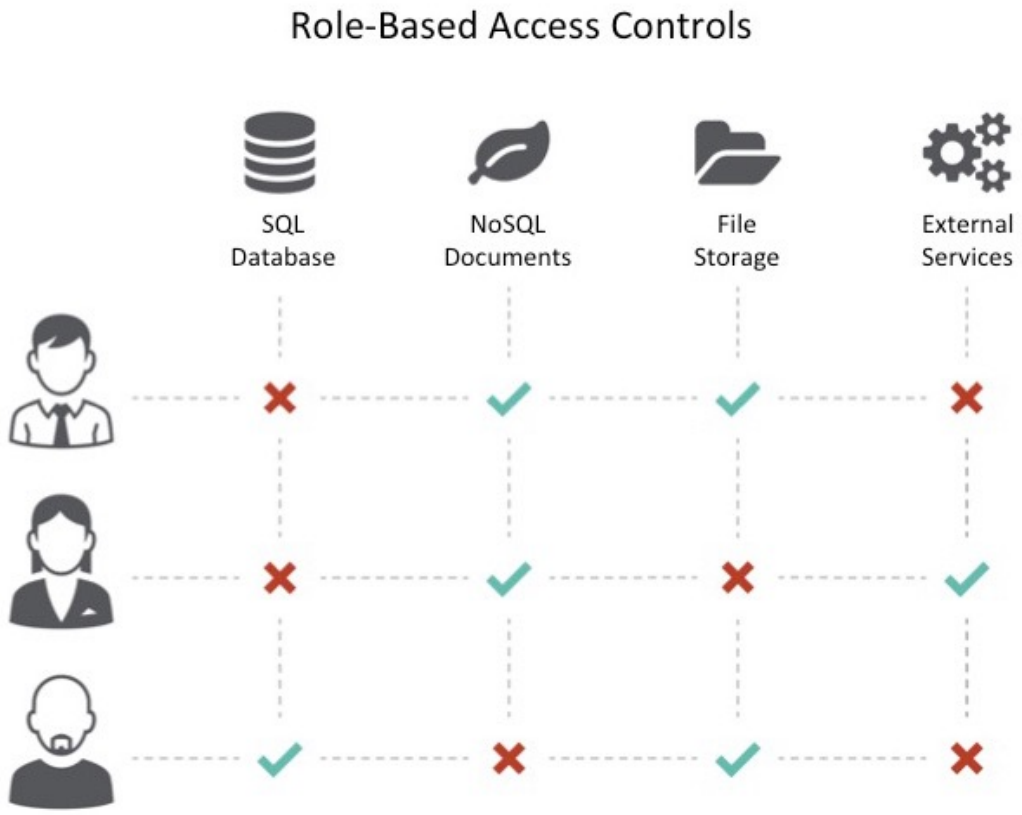
Applications written on a unified service platform also benefit from virtualization provided by the service layer. Services are no longer tied to specific pieces of backend infrastructure. Moving an application from development to testing and on to production is easy. The service platform can be installed anywhere, so applications can be moved between the cloud and data center. Application staging becomes simpler, and application lifecycle management becomes more secure.

Some DreamFactory services, such as database services, provide features such as aliasing to further abstract connected resources (i.e. database schema table and field names) represented in the API from the actual implementation.

Role-Based Access Controls

SQL databases have various tables, NoSQL databases have collections of documents, and file storage systems have different folders. In DreamFactory, administrators can define roles with role-based access controls to govern which of these components are visible to a given user, along with HTTP verbs such as GET, POST, PUT, PATCH or DELETE that govern access. You can use various combinations of these verbs and service resources to grant or deny user access.

Role-based access controls allow information to be hidden from certain roles and their assigned users or applications. For example, individual sales people might not have access to salary information, while this data might be available to managers. This capability prevents accidental data loss or disclosure of sensitive information.



Record-Level Access Control

Roles can be customized further with server-side filters to implement record-level access control. Each filter takes the form of a "field operator value" equation that must be true to enable access. The record-level access controls can impose constraints on external data sources. For example, you can limit data visibility only to records that the user created. This feature provides fine-grained security control of data sources.

Application API Keys

Applications in a DreamFactory instance can represent actual application code hosted on the instance, code hosted elsewhere on another server, or a designation for a native mobile app. Each application gets an API key generated at creation. The API key must be used for all access to the API (with a couple of exceptions noted below). The API key can be regenerated by administrative request, however care must be taken that all uses of the old key are updated (i.e. this is more difficult for a key embedded in a native mobile application).

Applications can be given “guest” access by assigning a default role to an application detailing what resources should be accessible without user authentication. This is useful, if your application has an open “website” or kiosk mode.

User Management

DreamFactory delivers extensive user management features. Single sign-on, user per-application roles, open registration, password resets, email services, email templates, and password hashing are all carefully implemented. The standardization of user management as a service and the integration of user management with the REST API platform prevent the security holes often found in custom implementations.

DreamFactory separates administrators from other users. Every DreamFactory instance must have at least one administrator configured at installation. Administrators can login, use the admin console, perform API calls, do not have roles, and are not required to use API keys for access. As such, these credentials should be safeguarded and usage minimized.

DreamFactory supports the standard username and password user management, but can also be provisioned to utilize most common OAuth providers and even LDAP and Active Directory.

Single Sign-On

When a user signs on to a DreamFactory instance via any of the aforementioned login options, they receive a session token, based on JSON Web Token (JWT) technology. The token is digitally signed and secures information about the identity of that user and the connection. This token is controlled by a configurable renewal and expiration policy and can be securely stored on the client. It must be included for all authenticated API usage.

Depending on the DreamFactory instance configuration, all network transactions can be conducted over HTTPS, thus adding the security capabilities of SSL/TLS to standard HTTP communications.

Deactivation of System Resources

Administrative updates to system resources are instantly reflected across all connections. Most system resources have an active/inactive state. When an administrator deactivates a user, application, role, or service, any further access to that resource is denied. For example, if a user or role is deactivated, any associated session tokens become invalid immediately. If an application is deactivated, any attempted access with that application's API key is denied. This provides a more secure environment at multiple levels of the API for events like people entering or leaving the company, service abuse, or temporary disablement of system resources.

Inheriting SQL Security

DreamFactory acts as a secure proxy for external data sources. When a SQL database is hooked up with a connection string, DreamFactory automatically inherits the security characteristics of that database. For example, if the connection string is for a read-only database user, then the REST API will deliver all the services as read-only, regardless of other access control settings.

DreamFactory can also inherit more complex permissions from a SQL database. You can configure server-side lookup keys to connect a user or role to the corresponding user or role on a remote database. This allows DreamFactory to mimic existing database security permissions, or change them as needed for mobile deployments.

Master Credential Storage

DreamFactory can connect to any number of external data sources and services. DreamFactory functions as a secure proxy that creates REST API services, provides server-side scripting, and enforces role-based access control. Each external data source or service will typically require a connection string, username, password, developer key, or some other type of master credential for access.

These master credentials are entered by an administrator in the DreamFactory Admin Console and encrypted for secure storage on the DreamFactory instance. When an end user logs in through single sign-on, they have controlled access to external data sources and services as enabled by their user role, but there is no way for them to discover the master credentials.

This capability to hide master credentials securely on the DreamFactory instance removes the need for client applications to use the master credentials for any external service. End user access is provided through single sign-on and limited by the user-base access controls. Another benefit of this architecture is that external services can be activated, deactivated, or redirected without changing the client software.

Programmable CORS access

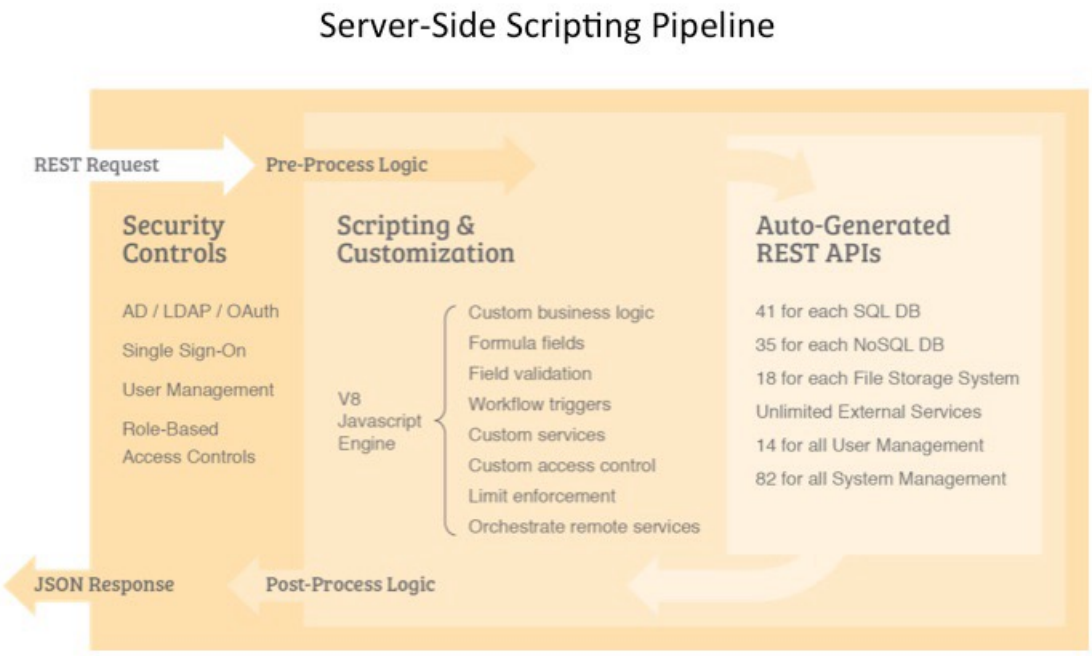
DreamFactory implements Cross-Origin Resource Sharing (CORS) as a system level web service. The Admin Panel has a simple interface that can enable any host domain to use the DreamFactory REST API. By default, CORS is turned off and the services are only available from the originating host. Programmable CORS support prevents cross-site scripting attacks and use of the API from unauthorized sources, but still allows the administrator to add necessary exceptions and temporary allowances for testing, etc.

Server-Side Scripting

DreamFactory uses the V8 Engine developed by Google to run server-side code written in JavaScript. Any request or response can be examined, modified, or rejected by the scripting environment. This enables developers to customize any API call or develop new services as needed. The V8 engine is sandboxed, so server-side scripts cannot interfere with other platform operations, operating system or hardware resources.

Scripting and customization can be used for formula fields, field validations, workflow triggers, access control, custom services, usage limits, and more. A third party can use server-side scripts to safely customize the backend system. Custom security protections and notifications can also be implemented with server-side scripts.

Server-side scripts can use the REST API when needed. For example, a script might call the email service, trigger a push notification, or store some information in the database. The role-based access controls have separate settings that govern data access for both client-side applications and server-side scripts. This capability enables server-side scripts to safely perform special operations that are not available from the client-side REST API.



SQL Injection Attacks

The DreamFactory REST API for SQL includes the ability to use query filter strings as URL parameters. For example, you could request all of the Opportunities greater than a certain amount, or created before a particular date. Each query filter string typically has a field name, operator, and target value. They can be combined into complex expressions with parentheses and logical operators.

A SQL injection attack can occur when a hacker types a tricky text string into a search field or constructs a malformed query filter for use in a REST API call. They might use strange escape sequences, incorrect type handling, or unusual patterns of fields, operators, and values.

DreamFactory deconstructs each query filter string into individual name, operator, and value components. The field name must match the object, the operator must make sense, and the field value must be well formed. After this, the query filter string is reconstructed with only valid parameters. This prevents unauthorized SQL statements from being injected into the database.

URL Routing Attacks

Another attack vector concerns spoofed URL strings, illegal URL parameters, or tricky HTTP headers that end up somehow bypassing role-based access controls or otherwise returning sensitive data from the platform. DreamFactory uses the routing engine in the Laravel PHP framework to avoid this problem.

The security of the REST API interface has also been rigorously tested by thousands of DreamFactory developers actively building projects. The fact that all of the REST API calls have a single entry point is also more secure than having separate entry points for each backend system.

Laravel Framework

DreamFactory is written with the latest version of PHP and the Laravel framework. PHP is the world's most widely used server-side programming language, and Laravel is the fastest growing framework for PHP. There is a very large community of professional developers using these products, and many mission-critical websites depend on them for secure implementations. DreamFactory builds upon several widely used Laravel extensions for things like user management, OAuth, LDAP, and URL routing for reliable and secure operation.

In Conclusion

DreamFactory is designed to be a secure platform for providing REST API services. Some of these capabilities are an inherent part of our open source model. Other features were specifically engineered to mobilize backend data sources in a secure manner. Reach out to the DreamFactory engineering team if you have additional questions.

Community Forum

Bugs and Feature Requests

Contact Support