

Automating Background Checks: Integrating Checkr API with Salesforce

In today's fast-paced hiring world, speed and accuracy are everything! 🌟 But when your agents are stuck doing double data entry—once in Salesforce and again in Checkr—it's not just a pain 😞, it's a productivity killer. So, we decided to automate the entire background check process using Flows, Apex, Triggers, and the Checkr RESTful API.

Here's how we made the magic happen... ✨🌟

What is Checkr?

Checkr is like your background check superhero 🦱—a web-based platform that helps businesses screen potential employees quickly and accurately. With its AI-powered system, Checkr automates searches for:



- ✅ Criminal records
- ✅ Employment & education verification
- ✅ Driving history
- ✅ Drug testing

It's all about making the hiring process faster, smarter, and compliant with legal regulations.



The Problem: Double Data Entry Drama

Before automation, our agents had to:

1. Fill candidate details in Salesforce 
2. Manually re-enter the SAME data in Checkr's dashboard 

This not only wasted time ⌚ but also increased the chance of errors ✖️.

We needed a solution—a way to automatically push data from Salesforce → Checkr without anyone lifting a finger! 📄💣

🔧 The Solution: Automating the Checkr Integration

We built a fully automated system that kicks off background checks with just a field update that matches the condition or a button click. No more manual work—just smooth sailing. 🚢

💻 Technologies We Used:

- Salesforce Flows – For no-code automation ⚙️
- Apex & Triggers – To handle complex logic 🔥
- RESTful API – For seamless communication with Checkr 🔗

🏠 How the Automation Works

We created two ways to trigger the background check:

📄 Automatic Trigger via Field Update that meets the criteria

When a specific field on the Account object is updated (e.g., “Background Check Required”), a trigger fires:

→ It grabs the Contact’s ID and calls an Apex method to send their info to Checkr.

🔗 Manual Trigger via Button

We also added a button on the Contact page layout.

When an agent clicks it, a Screen Flow launches, which:

- Collects the record's ID
- Calls the same Apex method to push data to Checkr

Simple, right? 🤖

📁 Checkr's Background Check Architecture

To start a background check in Checkr, you need to:


1. Create a Candidate – Send basic info (like email) to Checkr 📧
2. Create an Invitation – Trigger an email from Checkr requesting documents 📄

Once the candidate submits their documents, Checkr gets to work and generates a background check report. 🕵️♂️ 📄




🧠 Our Checkr Integration Workflow

1. Field Change or Button Click triggers our Apex method.
2. We call the Checkr API to create a candidate → send an invitation.
3. If anything fails, we:
 - Create a Task for the Account owner with an error message 🔴
 - Show the error instantly if using the button 🚨



Handling Checkr Webhooks: Real-Time Updates

Checkr doesn't leave us hanging—they send real-time updates via webhooks! 

We track key events like:



-  Background check cleared
-  Needs further review
-  Invitation canceled


When a webhook arrives, we:

- Update candidate status in Salesforce 
- Assign tasks for manual follow-ups if needed 

Custom Metadata for Dynamic Status Mapping

Checkr's webhook sends four key fields. To manage status labels, we:

1. Built a Custom Metadata Type to map field combinations 
2. Dynamically update statuses based on webhook responses 

No hardcoding—fully flexible! 

The Results: No More Double Data Entry!

Our Checkr-Salesforce integration:

- ✅ Eliminates manual data entry – No more double work!
- ✅ Improves accuracy – Less human error.
- ✅ Saves time – Fast, automated checks.
- ✅ Keeps everyone informed – Real-time updates with webhooks.

Automation FTW! 🎉

🔴 Final Thoughts

Integrating Checkr with Salesforce wasn't just a technical win—it was a game-changer for our hiring workflow. 🚀

By combining Flows, Apex, Triggers, and REST APIs, we transformed a manual, error-prone process into a fully automated system.

Why do things twice when you can automate it once? 😊